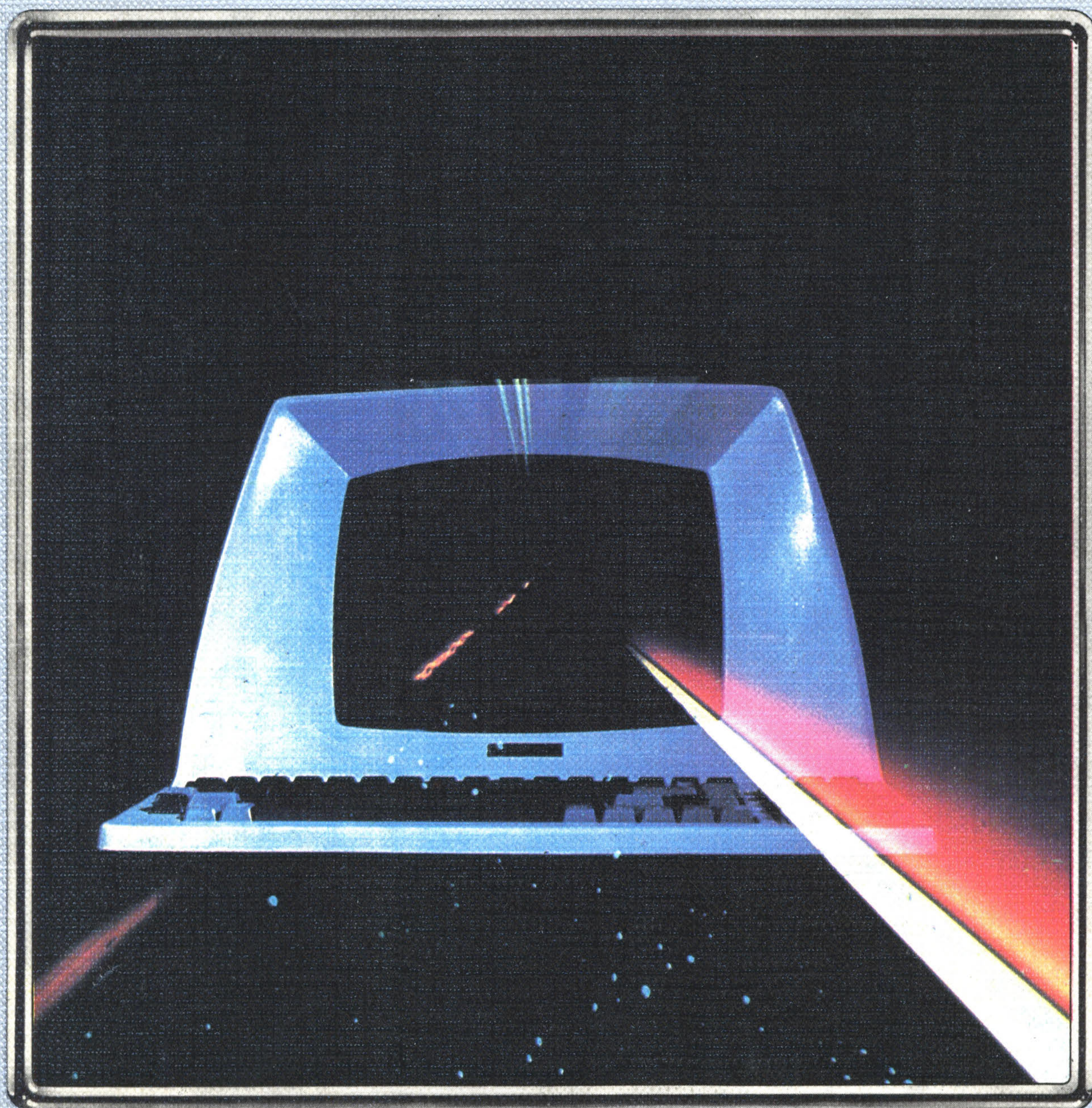
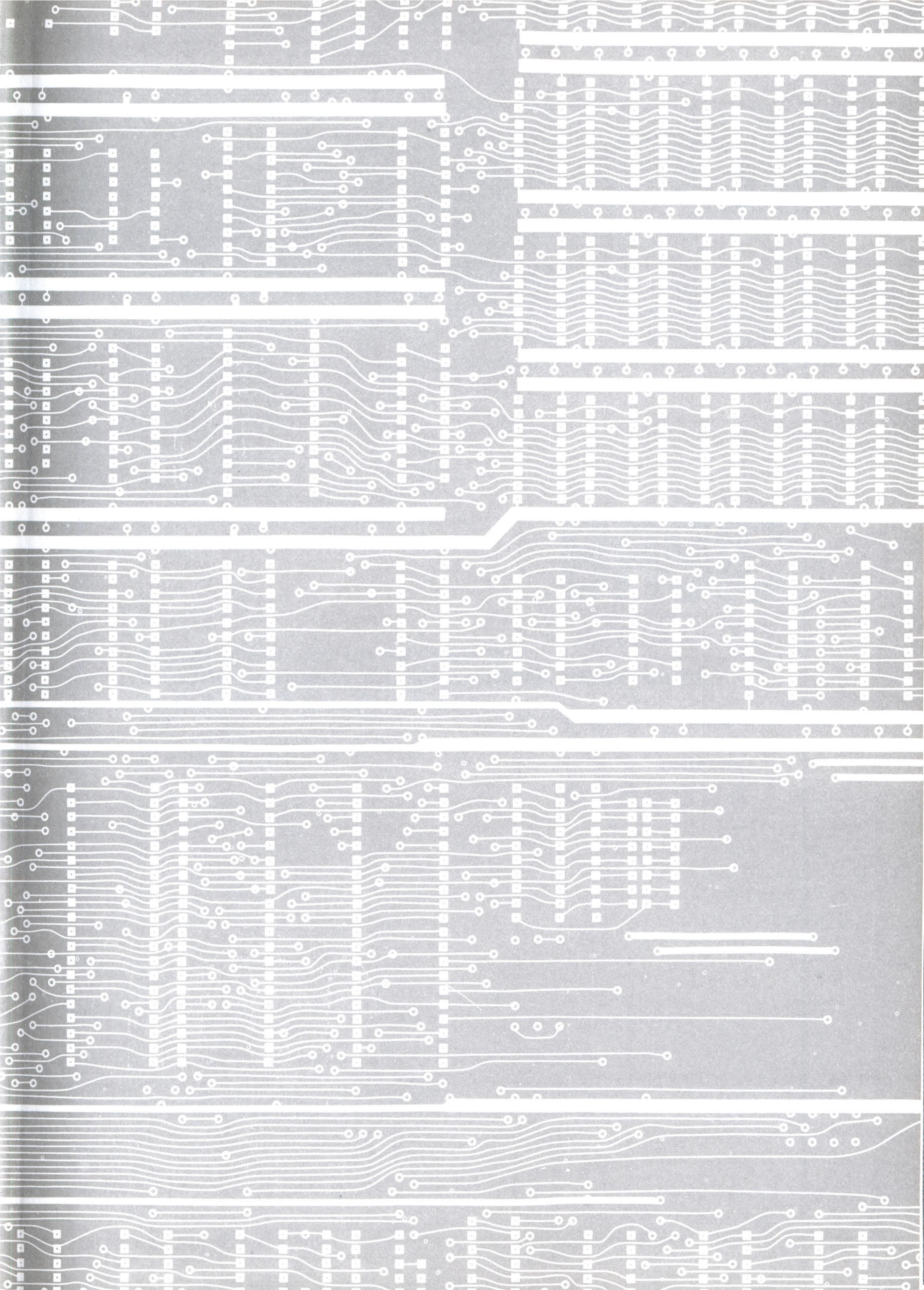
BASIC

ENCICLOPEDIA DE LA INFORMATICA
MINIORDENADORES Y ORDENADORES PERSONALES



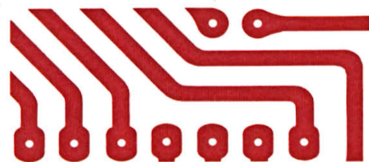
EDICIONES FORUM



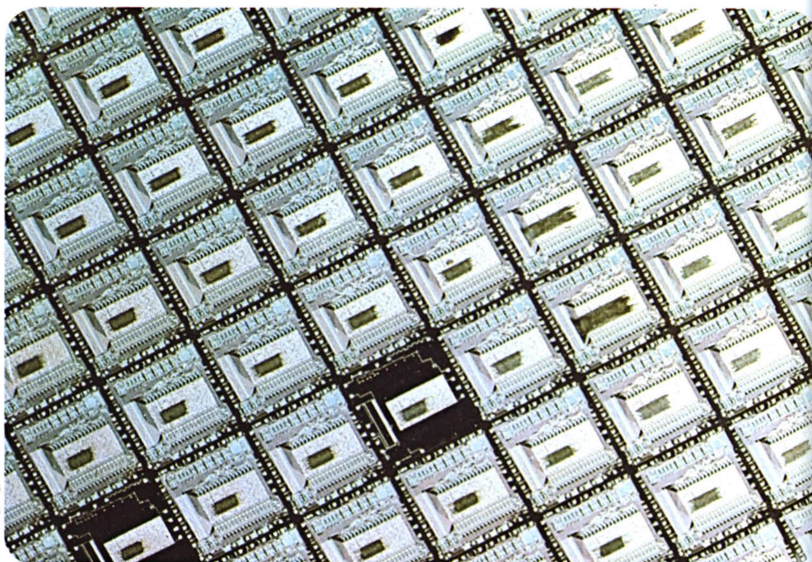


BASIC ADVANCED

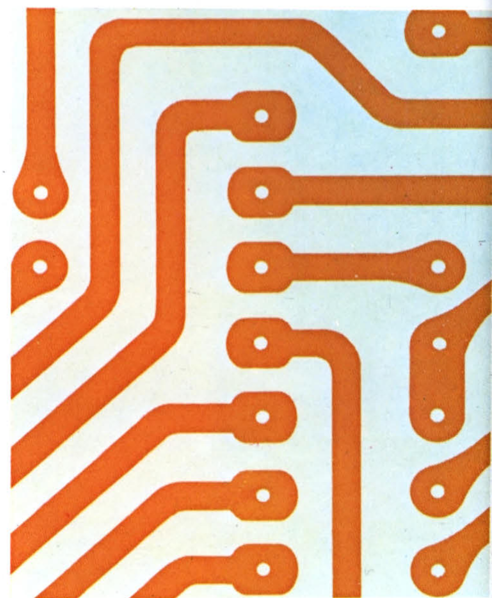
ENCICLOPEDIA DE LA INFORMATICA.
MINIORDENADORES Y ORDENADORES PERSONALES.
EL SOFTWARE DE APLICACION.



ENCICLOPEDIA DE LA INFORMATICA. MINIORDENADORES Y ORDENADORES PERSONALES. EL SOFTWARE DE APLICACION.

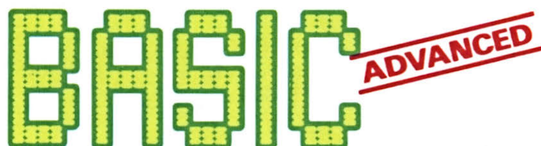

Marka


Marka-Leo de Wys



BASIC ADVANCED 9





ENCICLOPEDIA DE LA INFORMATICA. MINIORDENADORES Y ORDENADORES PERSONALES. EL SOFTWARE DE APLICACION.

Presidente
José Manuel Lara

Director Ejecutivo
Jesús Domingo

Dirección editorial
R.B.A., Proyectos Editoriales, S.A.

Dirección
Sante Senni
Con el asesoramiento de la Sociedad **E.G.S.**

Analista-programador: Stefano Senni

Programadores: Giancarlo Petrosino, Giampiero Censori, Alessandro Fiori, Marco Nardelli

Texto: Sante Senni, Ugo Spezia

REDACCION

Dirección: Gabriella Costarelli
Redactor Jefe: Marcella Marcaccini
Redacción: Ugo Spezia
Coordinación: Maria Pierantozzi
Corrección: María Albergo, Anna Landi, Laura Salvini, Rita Tancredi
Ilustración: Carla Bertini, Rossella Pozza
Dirección de arte: Vittorio Antinori
Compaginación: Riccardo Catani
Ilustraciones: Laura Fazio, Alessandra Fonzo, Roberto Giorgini, Francesco Izzo, Gianni Mazzoleni
Archivo: Stefano Bechelloni, Ulla Näslundh

VERSION ESPAÑOLA

Dirección: Jaime Gavaldá
Colaboración: Ramón Tapias - Microelectrónica y Control, S.A. - COMMODORE - José M.ª Amorós - Rifé Informática
Traducción: Jaime Gavaldá
Diseño de cubierta: Neslé Soulé
Producción: Ricardo Prats

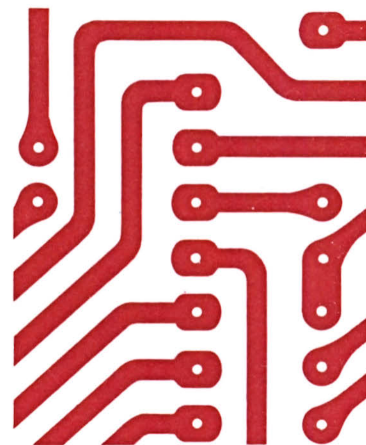
Composición electrónica: ITC-Fototipo. Barcelona-Madrid.

Imprime: CAYFOSA - Carretera de Caldas, Km. 3,7
Santa Perpetua de la Mogoda (Barcelona).

Depósito legal: B.37.099/83
ISBN (Obra completa): 84-7574-040-5
ISBN (Volumen IX): 84-7574-464-8
ISBN (Fascículos): 84-7574-044-8
Impreso en España - Printed in Spain.
Distribuye: Marco Ibérica, Distribuidora de Ediciones S.A.
Carretera de Irún, Km. 13,350, variante Fuencarral, 28034 Madrid.

El editor agradece colaboración de:

Alfa Romeo, Buffeti Data, Commodore Italiana, CPT Italia, Creazioni Walt Disney, Data General, Digital, Doxa, Elsag, Ericson, Facit Data Products, Ferrari, FIAT, Harden Italia, Hengstler Italia, Hewlett Packard, IBM, Intema, IRET Informática, Italcable, Lancia, Litton BEI, MEE, MSI Data Italia, Olivetti, Perkin-Elmer, Plessey Trading, Prime Italia, Rank Xerox, Rhône-Poulenc Italia, Sanco Ibex Italia, Sarin, Selca Elettronica, Selenia, Sperry, SIP, Telespazio y de Appel Computer S.p.A. y E.G.S. en la realización de esta obra.





Main

```
10 REM *********************
20 REM *    MAIN GENERAL    *
30 REM *********************
31 POKE650,0:PRINTCHR$(142)CHR$(8);:PRINT"♥";
32 PRINT"▒    E.G.S.           EDICIONES FORUM, S.A.  "
33 PRINT"▒                    ▀▀▀▀▀▀▀▀▀▀▀                ";
34 PRINT"▒                    ▀ LOS TESTS ▒              ";
35 PRINT"▒                    ▀▀▀▀▀▀▀▀▀▀▀               ";
40 PRINT"▒▒▒INTRODUCIR EL NUMERO CORRESPONDIENTE"
42 PRINT"▒▒▒▒▒▒▒▒▒AL PROGRAMA DESEADO"
44 PRINT"▒▒▒▒▒▒<1>▌ GENERACION TEST-PUNTOS"
45 PRINT"▒▒▒▒▒▒<2>▌ DESARROLLO TEST-PUNTOS"
46 PRINT"▒▒▒▒▒▒<3>▌ GENERACION TEST-ATRIBUTOS"
47 PRINT"▒▒▒▒▒▒<4>▌ DESARROLLO TEST-ATRIBUTOS"
48 PRINT"▒▒▒▒▒▒<5>▌ GENERACION TEST-SIMBOLOS"
49 PRINT"▒▒▒▒▒▒<6>▌ DESARROLLO TEST-SIMBOLOS"
50 PRINT"▒▒▒▒▒▒<0>▌ FIN DE TAREA"
51 INPUT"▒▒▒CUAL  :  ▌";R
52 IF R=0 THEN PRINT"♥":END
55 IF R<1 OR R>6 THEN 40
57 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒"SPC(4)"ESPERAR LA CARGA, POR FAVOR"
60 ON R GOTO 100,200,300,400,500,600
100 PRINT"▒POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
110 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"GP"CHR$(34)",8"
120 GOTO1000
200 PRINT"▒POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
210 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"UP"CHR$(34)",8"
220 GOTO1000
300 PRINT"▒POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
310 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"GA"CHR$(34)",8"
320 GOTO1000
400 PRINT"▒POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
410 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"UA"CHR$(34)",8"
420 GOTO 1000
500 PRINT"▒POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
510 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"GS"CHR$(34)",8"
520 GOTO 1000
600 PRINT"▒POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
610 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"US"CHR$(34)",8"
1000 POKE198,10:FORL=631TO640:READN:POKEL,N:NEXT
1010 DATA19,13,145,145,13,82,85,78,58,13

READY.
```

GP

```
12 REM ************* MAIN *************
13 PRINT"♥"
15 X=6:Y=11:GOSUB11000:PRINT"▒INTRODUCIR EL DISCO DATOS":Y=Y+3:X=X+4
16 GOSUB11000:PRINT"Y PULSAR UNA TECLA"
17 GET A$:IF A$<>"" THEN PRINTHO$:GOTO20
18 FOR T=1 TO 400:NEXTT:GOTO13
20 DIM B$(25),TP(25),LC(25),ND(25),CX(25),CY(25),DX(15),DY(15)
21 DIM TF(15),D$(15),H$(12),P$(20)
24 BL$=" ":PD$=".":AS$="-":NS$="#":SU$="▌":GIU$="▒":HO$="♥":SL$="/"
25 POKE 650,128
26 PRINTHO$
27 REM *** LECTURAS DE SERVICIO ***
30 RESTORE
35 FOR I=1 TO 15:READ TF(I):NEXTI
40 FOR I=1 TO 12:READ H$(I):NEXTI
45 REM
100 PRINTHO$:OZ=1:GOSUB 7000
110 X=2:Y=10:GOSUB11000:PRINT"▒NTRODUCIR LA SIGLA "
115 Y=Y+2:GOSUB11000:PRINT"▒DEL TEST (2 CARACTERES)▒ --▌":X=26
117 S$=""
119 Y=Y+1:GOSUB11000:PRINT"▒↑":Y=Y-1
120 FOR I=1 TO 2
122 GET L$:IF L$="" THEN 122
123 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
124 GOSUB11000:PRINT "▌"L$:X=X+1
126 Y=Y+1:GOSUB11000:PRINT"▒↑":Y=Y-1
132 S$=S$+L$
135 NEXT I
140 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
145 X=2:Y=15:GOSUB11000:PRINT"▒CUANTAS PREGUNTAS (1-20) ";:INPUT"▌";NL
150 IF NL<=0 OR NL>20 THEN GOSUB 9150:GOTO145
152 NF$="FD"+S$:GOSUB 6900:IF A=62 THEN 154
153 GOSUB 9006:RUN20
154 LR=35:GOSUB 12000
155 GOSUB6000:GOSUB6200
157 OZ=2:PRINTHO$:GOSUB 7030
160 IN=CY(2)
165 FOR K=2 TO 21:TP(K)=TP(2):LC(K)=LC(2):ND(K)=ND(2):CX(K)=CX(2)
170 IN=IN+1:CY(K)=IN:NEXT K
180 NUC=21
190 FOR LO=1 TO NL
195 B$(1)=STR$(LO)
200 GOSUB 2000
205 J=1
210 GOSUB 3040
215 REM CONTROLES COMANDO
220 IF FS=6 THEN 200
222 IF FS=7 THEN RUN20
225 IF FS=9 THEN 210
230 REM TRANSFIERE AL FICHERO FD+S$
235 J9=2+20*(LO-1):B=1
240 FOR K=J9 TO J9+19
243 B=B+1
245 PA=K:GOSUB 6800:PRINT#2,B$(B):INPUT#1,A,B$,C,D
247 NEXT K
250 NEXT LO
255 IF NL<10 THEN AA$="0"+RIGHT$(STR$(NL),1):GOTO 265
260 AA$=RIGHT$(STR$(NL),2)
265 FOR K9=1 TO 33:AA$=AA$+"#":NEXT K9
270 PA=1:GOSUB6800:PRINT#2,AA$
275 GOSUB 6600:GOSUB6400
280 REM INTRODUCCION RESPUESTAS
285 NF$="FR"+S$:GOSUB 6900:IF A=62 THEN 295
290 GOSUB 9006:RUN20
295 LR=35:GOSUB 12000
297 GOSUB6000:GOSUB6200
300 OZ=3:PRINTHO$:GOSUB 7000
305 IN=CY(2)
310 FOR J9=4 TO 20 STEP2
312 TP(J9)=TP(2):TP(J9+1)=TP(3):LC(J9)=LC(2):LC(J9+1)=LC(3):ND(J9)=ND(2)
314 ND(J9+1)=ND(3):CX(J9)=CX(2):CX(J9+1)=CX(3)
315 IN=IN+1:CY(J9)=IN:CY(J9+1)=IN
316 NEXT J9
318 NUC=21
320 FOR LO=1 TO NL
322 B$(1)=STR$(LO)
325 GOSUB 2000
327 J=1
330 GOSUB 3040
335 IF FS=6 THEN 325
340 IF FS=9 OR FS=7 THEN 330
345 J9=1+10*(LO-1):B=1
350 FOR K=J9 TO J9+9
352 B=B+1
354 AA$=B$(B)+B$(B+1)
356 PA=K:GOSUB 6800:PRINT#2,AA$:INPUT#1,A,B$,C,D
358 B=B+1
360 NEXT K
365 NEXT LO
370 GOSUB 6600:GOSUB6400
380 REM  INTRODUCCION GAMAS Y RESULTADO
385 NF$="FT"+S$:GOSUB 6900:IF A=62 THEN 395
390 GOSUB 9006:RUN20
395 LR=80:GOSUB 12000
400 GOSUB6000:GOSUB6200
405 OZ=4:PRINTHO$:GOSUB 7000
```

```
410 ID=DY(4)
411 FOR K=5 TO 13:D$(K)=STR$(K-3)+".":DX(K)=DX(4):ID=ID+1:DY(K)=ID
412 IF VAL(D$(K))=10 THEN DX(K)=DX(K)-1
413 NEXT K
414 IN=CY(1)
415 FOR K=3 TO 19 STEP 2
416 TP(K)=TP(1):LC(K)=LC(1):ND(K)=0:TP(K+1)=TP(1):LC(K+1)=LC(1):ND(K+1)=0
418 CX(K)=CX(1):CX(K+1)=CX(2):IN=IN+1:CY(K)=IN:CY(K+1)=IN
420 NEXT K
425 NUC=20:DS=13
430 GOSUB 2000
432 J=1
435 GOSUB 3040
440 IF FS=6 THEN 430
441 AA$=""
442 IF FS=7 OR FS=9 THEN 435
445 FOR LO=1 TO 20
447 V=VAL(B$(LO))
450 IF V<10 THEN B$(LO)="0"+RIGHT$(STR$(V),1)
452 AA$=AA$+B$(LO):P$(LO)=B$(LO)
455 NEXT LO
460 FOR I=1 TO 40:AA$=AA$+"#":NEXTI
465 PA=1:GOSUB 6800:PRINT#2,AA$:INPUT#1,A,B$,C,D
470 OZ=5:PRINTHO$:GOSUB7000
472 P=1
475 FOR LO=2 TO 11
480 B$(1)=P$(P):B$(2)=P$(P+1)
485 GOSUB 2000
490 J=1
495 GOSUB 3040
500 IF FS=6 THEN 485
505 IF FS=9 OR FS=7 THEN 495
510 P=P+2
515 AA$=B$(3)+B$(4)+B$(5)+B$(6)
520 PA=LO:GOSUB 6800:PRINT#2,AA$:INPUT#1,A,B$,C,D
525 NEXT LO
527 GOSUB6600:GOSUB6400
530 REM ****** FIN INTRODUCCIONES ******
550 OZ=1:PRINTHO$:GOSUB7000
555 X=3:Y=10:GOSUB11000:PRINT"▓CONTINUO (SI/NO) ";:INPUT "▐";R$
560 IF R$="NO" OR R$="/Γ" THENPRINTHO$:POKE646,6:CLR:LOAD"TEST",8
565 IF R$<>"SI" AND R$<>"♠↖" THEN 555
570 RUN
2000 REM
2005 REM *** PRESENTA MASCARA ***
2015 FORI=1TODS:X=DX(I):Y=DY(I):GOSUB11000:PRINT"◣"D$(I):NEXTI
2030 FOR J=1 TO NUC
2032 A$=""
2050 IF TP(J)<>5 AND TP(J)<>4 THEN 2060
2054 X=CX(J):Y=CY(J):GOSUB11000:PRINT"◣";B$(J)
2056 GOTO 2130
2060 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J)=2 THEN C$=NS$:GOTO2070
2064 C$=AS$
2070 FOR I=1 TO N1-N2-1:A$=A$+C$:NEXTI
2075 IF N2=0 THEN A$=A$+C$:GOTO 2095
2080 A$=A$+PD$
2090 FOR I=1 TO N2:A$=A$+"0":NEXTI
2095 X=CX(J):Y=CY(J):GOSUB 11000
2096 PRINT "▣";A$;"◣"
2097 B$(J)=""
2098 FOR K=1 TO N1-N2-1
2100 B$(J)=B$(J)+NS$
2105 NEXT K
2106 IF N2=0 THEN B$(J)=B$(J)+NS$:GOTO 2130
2110 B$(J)=B$(J)+PD$
2120 FORK=1TON2:B$(J)=B$(J)+NS$:NEXTK
2130 NEXT J
2150 RETURN
3000 REM
3010 REM ** DATA ENTRY **
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3065 IF FS=9 THEN RETURN
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 REM CARACTER
3105 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 FS=0:J=NUC:GOTO 3040
3150 IF J<NUC THEN J=J+1:GOTO 3040
3160 FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB11000:GOTO 3060
3200 FS=0:GOTO 3040
3210 REM
3220 IF FS>8 THEN FS=0:GOTO 3060
3230 IF FS=8 THEN FS=0:GOTO 3100
3235 RETURN
3240 REM
3250 REM  ** LECTURA DE 1 CAMPO **
3280 FS=0
3290 IF TP(NC)=4 THEN RETURN
3300 IF TP(NC)=2 THEN GOSUB 9000:RETURN
3310 IF MID$(B$(NC),CC,1)=PD$ THENX=X+S:CC=CC+S
3320 GOSUB 11000
3330 GET A$:IF A$<>"" THEN 3338
3332 GOSUB11000:PRINT"▀▄";:GOSUB11000:PRINT"◣";MID$(B$(NC),CC,1);
3336 GOTO 3330
3338 IFASC(A$)=19ORASC(A$)=20ORASC(A$)=147ORASC(A$)=148 ORASC(A$)=44  THEN 3330
3340 PRINT"▌▐";A$;
3350 CA=ASC(A$)
3360 FOR I=1 TO 15
3370 IF TF(I)=CA THEN FS=I:I=15:GOSUB3630:RETURN
3375 IF FS=9 THEN RETURN
3380 NEXTI
3390 IF TP(NC)<>1 THEN 3450
3400 IF CA=45 OR CA=46 OR(CA>=48ANDCA<=57)THEN3420
3410 GOTO 3330
3420 IF CA<>46 THEN3450
3430 GOSUB 4270
3440 IF SWP=1 THEN 3410
3450 S=1:GOSUB 3520
3460 IF CC=LC(NC) THEN RETURN
3470 CC=CC+1:X=X+1:GOTO 3310
3520 REM
3521 REM ** TRASFERENCIA AL BUFFER **
3540 REM
3570 IF CC=1 THEN SN$="":GOTO 3590
3580 SN$=LEFT$(B$(NC),CC-1)
3590 IF CC=LC(NC) THEN DE$="":GOTO3610
3600 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3610 B$(NC)=SN$+A$+DE$
3620 RETURN
3625 REM
3630 REM ** COMANDOS **
3640 REM
3680 IF FS>5 THEN 3710
3700 RETURN
3710 IF FS=9 THEN GOSUB 13000:RETURN
3720 IF FS=8 THEN GOSUB 4080:RETURN
3725 IF FS<8 THEN RETURN
3730 ON FS-9 GOSUB 3740,3910
3735 RETURN
3740 REM
3745 REM ** INSERCION **
3750 REM
3770 IF CC=LC(NC)THEN RETURN
3780 IF CC=1 THEN SN$="":GOTO3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DE$=RIGHT$(B$(NC),N)
3820 C$=BL$
3830 A$=SN$+C$+DE$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 REM
3890 PRINT B$(NC):S=-1
3900 IF RIGHT$(B$(NC),1)=PD$ THEN 3780
3905 RETURN
3909 REM
3910 REM ***** BORRADO *****
```

```
3920 REM
3930 IF TP(NC)<>1 THEN 3950
3940 IF MID$(B$(NC),2,1)=PD$ AND CC=1 THEN RETURN
3950 IF CC<1 THEN RETURN
3960 IF CC=LC(NC) THEN DE$="":GOTO 3980
3970 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3980 SN$=LEFT$(B$(NC),CC-1)
3990 IF TP(NC)=1 OR TP(NC)=2 THEN C$=NS$:GOTO 4010
4000 C$=AS$
4010 B$(NC)=SN$+DE$+C$
4020 XX=X:YY=Y
4030 X=CX(NC):Y=CY(NC):GOSUB 11000
4040 X=XX:Y=YY
4050 PRINT B$(NC):S=1:RETURN
4080 REM ******* ALINEADOS *******
4084 IF LC(NC)=1 THEN RETURN
4085 N1=LC(NC)-CC+1
4087 IF CC=1 THEN B$(NC)="":GOTO4100
4090 B$(NC)=LEFT$(B$(NC),CC-1)
4100 IF TP(NC)=1 OR TP(NC)=2 THEN 4150
4110 FOR I=1 TO N1
4120 B$(NC)=B$(NC)+BL$
4130 NEXT I
4140 GOTO  4190
4150 FOR I=1 TO N1
4160 B$(NC)=CHR$(32)+B$(NC)
4170 NEXT I
4180 IF RIGHT$(B$(NC),1)=PD$ THEN B$(NC)=BL$+LEFT$(B$(NC),LC(NC)-1)
4190 X=CX(NC):Y=CY(NC):GOSUB 11000
4210 PRINT B$(NC):RETURN
4270 REM *** CTRL PUNTO ***
4290 SWP=0
4300 IF CC=1 OR CC=LC(NC) THEN SWP=1:RETURN
4310 FOR I=1 TO LC(NC)
4320 IF MID$(B$(NC),I,1)=BL$ THEN I=LC(NC):SWP=1:RETURN
4330 NEXT I:RETURN
4340 REM
6000 REM
6001 REM *** GESTION DISCO ***
6010 REM -----------------------
6011 REM APERTURA CANAL COMANDOS
6012 REM -----------------------
6030 OPEN1,8,15
6035 RETURN
6200 REM ----------------
6204 REM APERTURA FICHERO
6206 REM ----------------
6210 OPEN2,8,3,NF$
6215 REM
6250 RETURN
6400 REM ---------------------
6402 REM CIERRE CANAL COMANDOS
6403 REM ---------------------
6410 CLOSE1
6415 RETURN
6490 REM
6600 REM --------------
6602 REM CIERRA FICHERO
6605 REM --------------
6610 CLOSE2
6620 RETURN
6800 REM ------------------
6801 REM APUNTA AL REGISTRO
6802 REM ------------------
6805 R1=PA:R2=0
6810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6815 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6820 RETURN
6825 REM
6900 REM CONTROLES EN LOS FICHEROS
6902 GOSUB6000:GOSUB6200
6904 INPUT#1,A,B$,C,D
6906 GOSUB6400:GOSUB6600
6910 RETURN
7000 REM
7001 REM *** CARGA LOS DATA ***
7002 VO$="    ENERACION  I -PUNTUACIONES"
7003 L=LEN(VO$)-3:L$=RIGHT$(VO$,L)
7004 U$="xOZ  1"
7005 P=(30-L)/2:P$="":FORK=1TOP:P$=P$+" ":NEXT K
7006 PRINT"";:PRINTCHR$(14)
7008 PRINT"          .I.   DICIONES  ORUM,  . ."
7011 PRINT"    ";U$;"                    "
7012 PRINT"          ";P$;
7015 PRINT""L$;"";P$
7020 PRINT"                    "
7030 X=1:Y=22:GOSUB11000:PRINT"────────────────────────────────────────"
7033 X=25:Y=23:GOSUB11000:PRINT"      "
7034 REM
7040 ON OZ GOTO 7110,7042,7044,7046,7048
7042 LV=27:GOSUB20000:NUC=2:DS=1:GOTO7090
7044 LV=40:GOSUB20000:NUC=3:DS=3:GOTO7090
7046 LV=64:GOSUB20000:NUC=2:DS=4:GOTO7090
7048 LV=86:GOSUB20000:NUC=6:DS=3:GOTO7090
7090 FOR I=1 TO DS:READ D$(I),DX(I),DY(I):NEXTI
7095 FOR I=1 TO NUC:READ TP(I),LC(I),ND(I),CX(I),CY(I)
7097 REM
7100 NEXT I
7110 RETURN
9000 REM *** ERRORES FICHEROS ***
9002 REM
9005 PRINT"":X=5:Y=12:GOSUB11000:PRINT"    ,  ISCO FUERA DE LINEA":GOTO 9010
9006 PRINT"":X=5:Y=12:GOSUB11000:PRINT"    , EL FICHERO DATOS YA"
9007 Y=Y+2:GOSUB11000:PRINT"SE HA CREADO EN ESTE DISCO"
9010 X=6:Y=21:GOSUB11000:PRINT"PULSE UNA TECLA PARA VOLVER"
9015 POKE198,0:WAIT198,1
9020 POKE198,0:RETURN
9150 REM ***** INDICA ERROR *********
9152 X=10:Y=20:GOSUB11000:PRINT"              "
9154 FOR T=1 TO3000:NEXTT
9156 X=10:Y=20:GOSUB11000:PRINT"                    "
9160 RETURN
10000 REM ************ DATA ***********
10001 DATA 145,17,29,157,140,133,136,13,25,134,135,0,0,0,0
10005 REM
10007 DATA"                        "
10010 DATA"                        "
10012 DATA"                        "
10014 DATA"                        "
10016 DATA"                        "
10018 DATA"       1                "
10020 DATA"       3                "
10022 DATA"       5                "
10024 DATA"       7                "
10026 DATA"       8                "
10028 DATA"                        "
10030 DATA"                        "
10040 REM ********* OZ=2 *********
10042 DATA"NTRODUCIR LA PREGUNTA N.",6,0
10043 DATA4,3,0,31,0
10044 DATA3,35,0,3,1
10050 REM ********* OZ=3 *********
10052 DATA"ESPUESTA A LA PREGUNTA N.",6,7
10053 DATA"ESPUESTA",2,9,"UNTOS",34,9
10055 DATA4,3,0,32,7
10056 DATA3,33,0,2,11
10057 DATA1,2,0,36,11
10060 REM ********* OZ=4 **********
10061 DATA"NTRODUCIR LAS GAMAS DE PUNTUACION",3,7
10062 DATA"\INIMO",10,10,"\AXIMO",19,10," 1.",7,12
10063 DATA1,2,0,12,12
10064 DATA1,2,0,22,12
10070 REM ********* OZ=5 ***********
10071 DATA"UNTUACION :",7,7," -",23,7
10072 DATA"JUICIOS :",14,11
10073 DATA4,2,0,20,7
10074 DATA4,2,0,25,7
10075 DATA3,20,0,9,14
10076 DATA3,20,0,9,16
10077 DATA3,20,0,9,18
10078 DATA3,20,0,9,20
10100 REM -------------------------
10200 REM *****************************
11000 REM
11001 REM **** POSICIONADO EN X,Y ****
11005 POKE 211,X
11010 POKE 214,Y
11015 SYS58732
```

```
11020 RETURN
12000 REM ***** CREA LOS FICHEROS *****
12005 GOSUB 6000
12010 FI$=NF$
12015 LR=LR+2
12020 OPEN 2,8,2,"0:"+FI$+",L,"+CHR$(LR)
12025 IF LEFT$(NF$,2)="FR" OR LEFT$(NF$,2)="FT" THEN AA$="##":GOTO 12035
12030 AA$="01"
12035 FOR K9=1 TO LR-4:AA$=AA$+"#":NEXTK9
12040 PA=1:GOSUB6800:PRINT#2,AA$
12045 CLOSE2
12050 CLOSE1
12100 RETURN
13000 REM *** AYUDA ***
13005 CO=X:CV=Y
13010 X=1:Y=23
13015 GOSUB11000:PRINTH$(1);"      ";
13017 FOR HE=2 TO 12
13020 GETR$:IFR$=""THEN13020
13021 IF R$<>CHR$(13) THEN 13020
13022 GOSUB11000:PRINTH$(HE)
13030 NEXT HE
13035 GETR$:IFR$=""THEN13035
13036 IF R$<>CHR$(13) THEN 13035
13040 GOSUB11000:PRINT"                              ";
13045 X=CO:Y=CV
13050 RETURN
20000 REM RESTORE DATA
20005 RESTORE
20045 IF LV<=0 THEN RETURN
20050 FOR I=1 TO LV:READ LV$:NEXTI
20100 RETURN
```

UP

```
12 REM ************** MAIN **************
13 PRINT"♥"
15 X=7:Y=11:GOSUB11000:PRINT"INTRODUCIR EL DISCO DATOS":Y=Y+3:X=X+4
16 GOSUB11000:PRINT"Y PULSAR UNA TECLA"
17 GET A$:IF A$<>"" THEN PRINTHO$:GOTO20
18 FOR T=1 TO 400:NEXTT:GOTO13
20 DIM PU(10),MI(10),MX(10),B$(4)
24 POKE53280,14:POKE53281,6:HO$="♥"
25 POKE 650,128
26 PRINTHO$
100 GOSUB 7000
110 X=2:Y=10:GOSUB11000:PRINT"NTRODUCIR LA SIGLA "
115 Y=Y+2:GOSUB11000:PRINT"DEL TEST (2 CARACTERES) --":X=26
117 S$=""
119 Y=Y+1:GOSUB11000:PRINT"↑":Y=Y-1
120 FOR I=1 TO 2
122 GET L$:IF L$="" THEN 122
123 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
124 GOSUB11000:PRINT "L$:X=X+1
126 Y=Y+1:GOSUB11000:PRINT"↑":Y=Y-1
132 S$=S$+L$
135 NEXT I
137 Y=Y+1:GOSUB11000:PRINT" ":Y=Y+1
140 REMARKS
145 PP=0
152 NF$="FD"+S$:GOSUB 6900:IF A=0 THEN 154
153 PRINTHO$:GOSUB 9000:RUN20
154 GOSUB6000:GOSUB6200
155 PA=1:GOSUB6800:INPUT#2,AA$:MX=VAL(AA$):REM ** PRINCIPIO BUCLE PREGUNTAS **
156 I=2:RS=1:DO=1
157 PRINTHO$:X=10:Y=0:GOSUB11000:PRINT"REGUNTA ";DO:X=3:Y=2
160 FOR K9=I TO I+19
162 PA=K9:GOSUB6800:INPUT#2,AA$
165 Y=Y+1:GOSUB11000:PRINTAA$
170 NEXT K9
175 GOSUB6400:GOSUB6600
180 X=2:Y=23:GOSUB11000:PRINT"PULSE UNA TECLA AL ULTIMAR LA LECTURA"
190 POKE 198,0:WAIT 198,1
195 NF$="FR"+S$:GOSUB6900:IF A=0 THEN 200
197 PRINTHO$:GOSUB 9000:RUN20
200 GOSUB6000:GOSUB6200:REM PRINCIPIO BUCLE RESPUESTAS POSIBLES
201 PRINTHO$:GOSUB7000:X=2:Y=7:GOSUB11000:PRINT"OSIBLES RESPUESTAS A LA PREGUN
TA";DO;":"
202 L9=0:Y=9
205 FOR J9=RS TO RS+9
210 PA=J9:GOSUB6800:INPUT#2,AA$
212 PU(L9)=VAL(MID$(AA$,34,2))
215 Y=Y+1:GOSUB11000:PRINTL9;".";LEFT$(AA$,33)
217 L9=L9+1
220 NEXT J9
222 GOSUB6400:GOSUB6600
223 POKE 198,0
225 Y=21:X=5:GOSUB11000:PRINT "ESPUESTA (0-9) ?"
227 GET R$:IF R$="" THEN 227
228 R=VAL(R$)
230 IF R<0 OR R>9 THEN 225
233 X=23:GOSUB11000:PRINT""R
235 PP=PP+PU(R)
236 X=5:Y=23:GOSUB11000:PRINT "EGURO (S/N)    ?";
237 GET SK$:IF SK$="" THEN 237
238 IF SK$="♥" OR SK$="S" THEN 245
239 IF SK$<>"/" AND SK$<>"N" THEN 241
240 X=23:Y=21:GOSUB11000:PRINT"  ":X=5:Y=23:GOSUB11000:PRINT"
 ":GOTO225
241 GOTO 237
244 REM ---------------------------------------
245 DO=DO+1:IF DO>MX THEN 260
250 I=I+20:RS=RS+10:NF$="FD"+S$:GOSUB6000:GOSUB6200:GOTO 157
260 REM PRINCIPIO LECTURA GAMAS Y JUICIOS
265 NF$="FT"+S$:GOSUB6900:IF A=0 THEN 280
270 PRINTHO$:GOSUB 9000:RUN 20
280 GOSUB6000:GOSUB6200
282 PA=1:GOSUB6800:INPUT#2,AA$
284 J=1
286 FOR I=1 TO 40 STEP4
288 MI(J)=VAL(MID$(AA$,I,2))
290 MX(J)=VAL(MID$(AA$,I+2,2))
292 J=J+1
294 NEXT I
296 I9=0:I=1
300 IF PP>=MI(I) AND PP<=MX(I) THEN I9=1:GOTO 307
305 I=I+1:IF I<=10 THEN 300
307 IF I9<>0 THEN 330
320 PRINTHO$:X=2:Y=10:GOSUB11000:PRINT"  EN FASE DE GENERACION"
322 Y=Y+1:GOSUB11000:PRINT"SE HAN INTRODUCIDO DATOS ERRONEOS"
324 X=5:Y=20:GOSUB11000:PRINT"PULSAR UNA TECLA PARA CONTINUAR"
326 POKE 198,0:WAIT198,1:RUN
330 PA=I+1:GOSUB6800:INPUT#2,AA$
335 GOSUB6400:GOSUB6600
340 B$(1)=LEFT$(AA$,20):B$(2)=MID$(AA$,21,20)
345 B$(3)=MID$(AA$,41,20):B$(4)=RIGHT$(AA$,20)
350 PRINTHO$:GOSUB 7000:X=10:Y=9:GOSUB11000:PRINT"      :"
355 Y=Y+3:X=10
360 FOR J=1 TO 4:Y=Y+1:GOSUB11000:PRINTB$(J):NEXT J
530 REM ****** CONTINUA (SI-NO) ******
555 X=6:Y=21:GOSUB11000:PRINT"TRO TEST (SI/NO) ";:INPUT "";R$
560 IF R$="NO" OR R$="/" THENPRINTHO$:POKE646,6:CLR:LOAD"TEST",8
565 IF R$<>"SI" AND R$<>"♥" THEN 555
570 RUN20
575 REM ***************************************
6000 REM
6001 REM *** GESTION DISCO ***
6010 REM ----------------------
6011 REM APERTURA CANAL COMANDOS
6012 REM ----------------------
6030 OPEN1,8,15
6035 RETURN
6200 REM ---------------
6204 REM APERTURA FICHERO
6206 REM ---------------
6210 OPEN2,8,3,NF$
6215 REM
6250 RETURN
6400 REM ----------------------
6402 REM CIERRE CANAL COMANDOS
6403 REM ----------------------
6410 CLOSE1
6415 RETURN
6490 REM
6600 REM -------------
6602 REM CIERRA FICHERO
6605 REM -------------
```

```
6610 CLOSE2
6620 RETURN
6800 REM ------------------
6801 REM APUNTA AL REGISTRO
6802 REM ------------------
6805 R1=PA:R2=0
6810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6815 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6820 RETURN
6825 REM
6900 REM CONTROLES SOBRE LOS FICHEROS
6902 GOSUB6000:GOSUB6200
6904 INPUT#1,A,B$,C,D
6906 GOSUB6400:GOSUB6600
6910 RETURN
7000 REM
7001 REM *** CABECERA ***
7002 VO$="   ESARROLLO  CON PUNTUACIONES"
7003 L=LEN(VO$)-3:L$=RIGHT$(VO$,L)
7004 U$=" IEST "
7005 P=(30-L)/2:P$="":FORK=1TOP:P$=P$+" ":NEXT K
7006 PRINT"";:PRINTCHR$(14)
7008 PRINT"          .I.♥.    DICIONES  ORUM, ♥.♠."
7011 PRINT"  ";U$;"                "
7012 PRINT"          ";P$;
7015 PRINT""L$;"";P$
7020 PRINT"                                  "
7034 REM
7110 RETURN
9000 REM *** ERRORES FICHEROS ***
9002 IF A=62 THEN 9006
9005 X=5:Y=12:GOSUB11000:PRINT"  ,  ISCO FUERA DE LINEA":GOTO 9010
9006 X=5:Y=12:GOSUB11000:PRINT"  , EL FICHERO DATOS NO"
9007 Y=Y+2:GOSUB11000:PRINT"ESTA EN ESTE DISCO"
9010 X=6:Y=21:GOSUB11000:PRINT"PULSE UNA TECLA PARA VOLVER"
9015 POKE198,0:WAIT198,1
9020 POKE198,0:RETURN
11000 REM
11001 REM **** POSICIONADO EN X,Y ****
11005 POKE 211,X
11010 POKE 214,Y
11015 SYS58732
11020 RETURN
```

GA

```
12 REM ************** MAIN **************
13 PRINT""
15 X=7:Y=11:GOSUB11000:PRINT"INTRODUCIR EL DISCO DATOS":Y=Y+3:X=X+3
16 GOSUB11000:PRINT"Y PULSAR UNA TECLA"
17 GET A$:IF A$<>"" THEN PRINTHO$:GOTO20
18 FOR T=1 TO 400:NEXTT:GOTO13
20 DIM B$(25),TP(25),LC(25),ND(25),CX(25),CY(25),DX(15),DY(15)
21 DIM TF(15),D$(15),H$(12),PK$(21)
24 BL$=" ":PD$=".":AS$="-":NS$="#":SU$="":GIU$="":HO$="":SL$="/"
25 POKE 650,128
26 PRINTHO$
27 REM *** LECTURAS DE SERVICIO ***
30 RESTORE
35 FOR I=1 TO 15:READ TF(I):NEXTI
40 FOR I=1 TO 12:READ H$(I):NEXTI
45 REM
100 PRINTHO$:OZ=1:GOSUB 7000
110 X=2:Y=10:GOSUB11000:PRINT"NTRODUCIR LA SIGLA "
115 Y=Y+2:GOSUB11000:PRINT"DEL TEST (2 CARACTERES) --":X=26
117 S$=""
119 Y=Y+1:GOSUB11000:PRINT"↑":Y=Y-1
120 FOR I=1 TO 2
122 GET L$:IF L$="" THEN 122
123 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
124 GOSUB11000:PRINT ""L$:X=X+1
126 Y=Y+1:GOSUB11000:PRINT"↑":Y=Y-1
132 S$=S$+L$
135 NEXT I
140 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
152 NF$="AT"+S$:GOSUB 6900:IF A=62 THEN 154
153 GOTO 15000:REM *** VARIACION ***
154 LR=30:GOSUB 12000
155 GOSUB6000:GOSUB6200
157 OZ=2:PRINTHO$:VO$="   NTRODUCCION":GOSUB 7003
160 IN=CY(1)
165 FOR K=2 TO 10:TP(K)=TP(1):LC(K)=LC(1):ND(K)=ND(1):CX(K)=CX(1)
170 IN=IN+1:CY(K)=IN:NEXT K
180 NUC=10
200 GOSUB 2000
205 J=1
210 GOSUB 3040
215 REM CONTROLES COMANDO
220 IF FS=6 THEN 200
222 IF FS=7 THEN RUN20
225 IF FS=9 THEN 210
230 REM TRANSFIERE AL FICHERO AT+S$
235 FOR K9=1 TO 10
240 PA=K9:GOSUB6800:PRINT#2,B$(K9):INPUT#1,A,B$,C,D
245 NEXT K9
265 GOSUB6400:GOSUB6600
266 B=1:FOR K=2 TO 20STEP2:PK$(K)=B$(B):B=B+1:NEXT K
280 REM INTRODUCCION ELEMENTOS
285 NF$="EL"+S$
295 LR=30:GOSUB 12000:MX=1
297 GOSUB6000:GOSUB6200
300 OZ=3:PRINTHO$:VO$="   NTRODUCCION":GOSUB 7003
305 IN=CY(2):B=2
310 FOR K=2 TO 21STEP2
311 B$(K)=PK$(B)
312 TP(K)=TP(2):TP(K+1)=TP(3):LC(K)=LC(2):LC(K+1)=LC(3):ND(K)=ND(2)
314 ND(K+1)=ND(3):CX(K)=CX(2):CX(K+1)=CX(3)
315 IN=IN+1:CY(K)=IN:CY(K+1)=IN:B=B+2
316 NEXT K
318 NUC=21
325 GOSUB 2000
327 J=1
330 GOSUB 3040
335 IF FS=6 THEN 325
340 IF FS=9 OR FS=7 THEN 330
350 A$=""
355 FOR K=3 TO 21STEP2
360 A$=A$+B$(K)
365 NEXT K
370 AA$=B$(1)+A$
380 REM INTRODUCE EN EL FICHERO EL+S$
395 MX=MX+1:IF MX>100 THEN A=100:PRINTHO$:GOSUB 9000:RUN20
390 PA=MX:GOSUB 6800:PRINT#2,AA$
395 REM
400 GOSUB6400:GOSUB6600
530 REM ****** FIN INTRODUCCIONES ******
550 OZ=1:PRINTHO$:GOSUB7000
555 X=3:Y=10:GOSUB11000:PRINT"TRA INTRODUCCION (SI/NO) ";:INPUT "";R$
560 IF R$="NO" OR R$="/" THEN 600
565 IF R$<>"SI" AND R$<>"♥" THEN 555
570 NF$="EL"+S$:GOTO 297
600 REM -- FIN --
605 MX$=STR$(MX):IF MX<10 THEN MX$="00"+RIGHT$(MX$,1):GOTO 650
610 IF MX<100 THEN MX$="0"+RIGHT$(MX$,2):GOTO 650
620 MX$=RIGHT$(MX$,3)
650 NF$="EL"+S$:GOSUB6000:GOSUB6200
655 AA$=MX$
660 FOR K=1 TO 27:AA$=AA$+CHR$(35):NEXT K
665 PA=1:GOSUB6800:PRINT#2,AA$
670 GOSUB6400:GOSUB6600
675 PRINTHO$:POKE 646,6:CLR:LOAD"TEST",8
2000 REM
2005 REM *** PRESENTA MASCARA ***
2015 FORI=1TODS:X=DX(I):Y=DY(I):GOSUB11000:PRINT""D$(I):NEXTI
2030 FOR J=1 TO NUC
2032 A$=""
2050 IF TP(J)<>5 AND TP(J)<>4 THEN 2060
2054 X=CX(J):Y=CY(J):GOSUB11000:PRINT"";B$(J)
2056 GOTO 2130
2060 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J)=2 THEN C$=NS$:GOTO2065
2064 C$=AS$
2065 IF N1=1 THEN A$=C$:GOTO 2095
2070 FOR I=1 TO N1-N2-1:A$=A$+C$:NEXTI
2075 IF N2=0 THEN A$=A$+C$:GOTO 2095
2080 A$=A$+PD$
2090 FOR I=1 TO N2:A$=A$+"0":NEXTI
2095 X=CX(J):Y=CY(J):GOSUB 11000
2096 PRINT "";A$;""
```

```
2097 B$(J)=""
2098 FOR K=1 TO N1-N2-1
2100 B$(J)=B$(J)+NS$
2105 NEXT K
2106 IF N2=0 THEN B$(J)=B$(J)+NS$:GOTO 2130
2110 B$(J)=B$(J)+PD$
2120 FORK=1TON2:B$(J)=B$(J)+NS$:NEXTK
2130 NEXT J
2150 RETURN
3000 REM
3010 REM ** DATA ENTRY **
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3065 IF FS=9 THEN RETURN
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 REM CARACTER
3105 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 FS=0:J=NUC:GOTO 3040
3150 IF J<NUC THEN J=J+1:GOTO 3040
3160 FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB11000:GOTO 3060
3200 FS=0:GOTO 3040
3210 REM
3220 IF FS>8 THEN FS=0:GOTO 3060
3230 IF FS=8 THEN FS=0:GOTO 3100
3235 RETURN
3240 REM
3250 REM  ** LECTURA DE 1 CAMPO **
3280 FS=0
3290 IF TP(NC)=4 THEN RETURN
3300 IF TP(NC)=2 THEN GOSUB 9000:RETURN
3310 IF MID$(B$(NC),CC,1)=PD$ THENX=X+S:CC=CC+S
3320 GOSUB 11000
3330 GET A$:IF A$<>"" THEN 3338
3332 GOSUB11000:PRINT"▬";:GOSUB11000:PRINT"◣";MID$(B$(NC),CC,1);
3336 GOTO 3330
3338 IFASC(A$)=19ORASC(A$)=20ORASC(A$)=147ORASC(A$)=148 ORASC(A$)=44  THEN 3330
3340 PRINT"▮▮";A$;
3350 CA=ASC(A$)
3360 FOR I=1 TO 15
3370 IF TF(I)=CA THEN FS=I:I=15:GOSUB3630:RETURN
3375 IF FS=9 THEN RETURN
3380 NEXTI
3390 IF TP(NC)<>1 THEN 3450
3400 IF CA=45 OR CA=46 OR(CA>=48ANDCA<=57)THEN3420
3410 GOTO 3330
3420 IF CA<>46 THEN3450
3430 GOSUB 4270
3440 IF SWP=1 THEN 3410
3450 S=1:GOSUB 3520
3460 IF CC=LC(NC) THEN RETURN
3470 CC=CC+1:X=X+1:GOTO 3310
3520 REM
3521 REM ** TRASFERENCIA AL BUFFER **
3540 REM
3570 IF CC=1 THEN SN$="":GOTO 3590
3580 SN$=LEFT$(B$(NC),CC-1)
3590 IF CC=LC(NC) THEN DE$="":GOTO3610
3600 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3610 B$(NC)=SN$+A$+DE$
3620 RETURN
3625 REM
3630 REM ** COMANDOS **
3640 REM
3680 IF FS>5 THEN 3710
3700 RETURN
3710 IF FS=9 THEN GOSUB 13000:RETURN
3720 IF FS=8 THEN GOSUB 4080:RETURN
3725 IF FS<8 THEN RETURN
3730 ON FS-9 GOSUB 3740,3910
3735 RETURN
3740 REM
3745 REM ** INSERCION **
3750 REM
3770 IF CC=LC(NC)THEN RETURN
3780 IF CC=1 THEN SN$="":GOTO3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DE$=RIGHT$(B$(NC),N)
3820 C$=BL$
3830 A$=SN$+C$+DE$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 REM
3890 PRINT B$(NC):S=-1
3900 IF RIGHT$(B$(NC),1)=PD$ THEN 3780
3905 RETURN
3909 REM
3910 REM ***** BORRADO *****
3920 REM
3930 IF TP(NC)<>1 THEN 3950
3940 IF MID$(B$(NC),2,1)=PD$ AND CC=1 THEN RETURN
3950 IF CC<1 THEN RETURN
3960 IF CC=LC(NC) THEN DE$="":GOTO 3980
3970 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3980 SN$=LEFT$(B$(NC),CC-1)
3990 IF TP(NC)=1 OR TP(NC)=2 THEN C$=NS$:GOTO 4010
4000 C$=AS$
4010 B$(NC)=SN$+DE$+C$
4020 XX=X:YY=Y
4030 X=CX(NC):Y=CY(NC):GOSUB 11000
4040 X=XX:Y=YY
4050 PRINT B$(NC):S=1:RETURN
4080 REM ******* ALINEADOS *******
4084 IF LC(NC)=1 THEN RETURN
4085 N1=LC(NC)-CC+1
4087 IF CC=1 THEN B$(NC)="":GOTO4100
4090 B$(NC)=LEFT$(B$(NC),CC-1)
4100 IF TP(NC)=1 OR TP(NC)=2 THEN 4150
4110 FOR I=1 TO N1
4120 B$(NC)=B$(NC)+BL$
4130 NEXT I
4140 GOTO  4190
4150 FOR I=1 TO N1
4160 B$(NC)=CHR$(32)+B$(NC)
4170 NEXT I
4180 IF RIGHT$(B$(NC),1)=PD$ THEN B$(NC)=BL$+LEFT$(B$(NC),LC(NC)-1)
4190 X=CX(NC):Y=CY(NC):GOSUB 11000
4210 PRINT B$(NC):RETURN
4270 REM *** CTRL PUNTO ***
4290 SWP=0
4300 IF CC=1 OR CC=LC(NC) THEN SWP=1:RETURN
4310 FOR I=1 TO LC(NC)
4320 IF MID$(B$(NC),I,1)=BL$ THEN I=LC(NC):SWP=1:RETURN
4330 NEXT I:RETURN
4340 REM
6000 REM
6001 REM *** GESTION DISCO ***
6010 REM -----------------------
6011 REM APERTURA CANAL COMANDOS
6012 REM -----------------------
6030 OPEN1,8,15
6035 RETURN
6200 REM ---------------
6204 REM APERTURA FICHERO
6206 REM ---------------
6210 OPEN2,8,3,NF$
6215 REM
6250 RETURN
6400 REM --------------------
6402 REM CIERRE CANAL COMANDOS
6403 REM --------------------
6410 CLOSE1
6415 RETURN
6490 REM
6600 REM -------------
6602 REM CIERRA FICHERO
6605 REM -------------
```

```
6610 CLOSE2
6620 RETURN
6800 REM ------------------
6801 REM APUNTA AL REGISTRO
6802 REM ------------------
6805 R1=PA:R2=0
6810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6815 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6820 RETURN
6825 REM
6900 REM CONTROLES SOBRE LOS FICHEROS
6902 GOSUB6000:GOSUB6200
6904 INPUT#1,A,B$,C,D
6906 GOSUB6400:GOSUB6600
6910 RETURN
7000 REM
7001 REM *** CARGA LOS DATA ***
7002 VO$="   ⎸ENERACION - ATRIBUTOS"
7003 L=LEN(VO$)-3:L$=RIGHT$(VO$,L)
7004 U$=" ⎸EST "
7005 P=(30-L)/2:P$="":FORK=1TOP:P$=P$+" ":NEXT K
7006 PRINT"▒";:PRINTCHR$(14)
7008 PRINT"▮▧▨          ▩┌.⎸.♥.     ‾DICIONES _ORUM, ♥.♠."
7011 PRINT"▮▧▨ ";U$;" ▩                                  "
7012 PRINT"▮▧▨          ▩";P$;
7015 PRINT"▬▩"L$;"▩";P$
7020 PRINT"▮▮▮▮▮▮▮▮▩                               "
7025 IF OZ=1 THEN RETURN
7030 X=1:Y=22:GOSUB11000:PRINT"▨───────────────────────────────────"
7033 X=25:Y=23:GOSUB11000:PRINT"▲‾⎸_⊥+⎸ : ♠⎸/‾♠"
7034 REM
7040 ON OZ GOTO 7110,7042,7044
7042 LV=27:GOSUB20000:NUC=1:DS=1:GOTO7090
7044 LV=35:GOSUB20000:NUC=3:DS=2:GOTO7090
7090 FOR I=1 TO DS:READ D$(I),DX(I),DY(I):NEXTI
7095 FOR I=1 TO NUC:READ TP(I),LC(I),ND(I),CX(I),CY(I)
7097 REM
7100 NEXT I
7110 RETURN
9000 REM *** ERRORES FICHEROS ***
9002 IF A=100 THEN 9008
9005 X=5:Y=12:GOSUB11000:PRINT"▮┌──┌─ , ‾ISCO FUERA DE LINEA":GOTO 9010
9006 X=5:Y=12:GOSUB11000:PRINT"▮┌──┌─ , EL FICHERO DATOS YA"
9007 Y=Y+2:GOSUB11000:PRINT"SE HA CREADO EN ESTE DISCO":GOTO 9010
9008 X=5:Y=12:GOSUB11000:PRINT"▮┌──┌─ , ESPACIO DATOS AGOTADO"
9010 X=8:Y=21:GOSUB11000:PRINT"▮PULSE UNA TECLA PARA VOLVER"
9015 POKE198,0:WAIT198,1
9016 IF A=70 THEN POKE 198,0:RUN 20
9020 POKE198,0:RETURN
9150 REM ***** INDICA ERROR *********
9152 X=10:Y=20:GOSUB11000:PRINT"▮┌──┌─ ‾‾ ⟍/⎸┌‾ ⌒┌/"
9154 FOR T=1 TO3000:NEXTT
9156 X=10:Y=20:GOSUB11000:PRINT"                          "
9160 RETURN
10000 REM ************* DATA ************
10001 DATA 145,17,29,157,140,133,136,13,25,134,135,0,0,0,0
10005 REM
10007 DATA"▲   < _‾⎸ ⌐/ >        ▮⌐♠_♠ ‾⌐/⎸⟍/♠_"
10010 DATA"▲   ‾♥_ _⟍⎸ ⎸⎸  : ▮ ‾‾♥‾L♠♦♠ ♠ ‾‾_‾⎸♠"
10012 DATA"▲   ‾♥_ L‾_⎸   : ▮ ‾‾♥‾L♠♦♠ ♠ ⟍♦● ⌒‾_‾♠"
10014 DATA"▲   ‾♥_ ⌐⎸     : ▮ ‾‾♥‾L♠♦♠ ♠_⟍⎸♠     "
10016 DATA"▲   ‾♥_ ‾TO/   : ▮ ‾‾♥‾L♠♦♠ ♠⎸♠⎸‾  "
10018 DATA"▲       ´_1´   : ▮ ♠/⊥♠ ⟍/⎸┌‾ ⌒⟍┌/"
10020 DATA"▲       ´_3´   : ▮ ⟍/♥‾⎸♠ ‾♠_♠‾⎸‾_ "
10022 DATA"▲       ´_5´   : ▮ ⎸┌_♠ ‾♠_♠‾⎸‾_  "
10024 DATA"▲       ´_7´   : ▮ X/‾LX‾ ♠L ⟍‾/⁄  "
10026 DATA"▲       ´_8´   : ▮ ⟍/⎸┌‾ ⌒‾ ‾/ ⟍♥┌  "
10028 DATA"▲   ‾⎸_⊥+⎸     : ▮ ⟍‾‾ L♠ ♠⎸/‾♠  "
10030 DATA"▲_‾⎸ ⌐/        : ▮ ♠L⟍/‾♠  ⎸ ‾⌐/X♠L⟍‾♠"
10040 REM ********* OZ=2 *********
10042 DATA"⟍NTRODUCIR LOS ATRIBUTOS",7,8
10043 DATA3,30,0,5,11
10050 REM ********* OZ=3 *********
10052 DATA"‾LEMENTO : ",4,7
10053 DATA"ATRIBUTO",4,9
10055 DATA3,20,0,18,7
10056 DATA4,30,0,2,9
10057 DATA1,1,0,36,9
11000 REM
11001 REM **** POSICIONADO EN X,Y ****
11005 POKE 211,X
11010 POKE 214,Y
11015 SYS58732
11020 RETURN
12000 REM ***** CREA LOS FICHEROS *****
12005 GOSUB 6000
12010 FI$=NF$
12015 LR=LR+2
12020 OPEN 2,8,2,"0:"+FI$+",L,"+CHR$(LR)
12025 IF LEFT$(NF$,2)="AT" THEN AA$="##":GOTO 12035
12030 AA$="01"
12035 FOR K9=1 TO LR-4:AA$=AA$+"#":NEXTK9
12040 PA=1:GOSUB6800:PRINT#2,AA$
12045 CLOSE2
12050 CLOSE1
12100 RETURN
13000 REM *** AYUDA ***
13005 CO=X:CV=Y
13010 X=1:Y=23
13015 GOSUB11000:PRINTH$(1);"       ";
13017 FOR HE=2 TO 12
13020 GETR$:IFR$=""THEN13020
13021 IF R$<>CHR$(13) THEN 13020
13022 GOSUB11000:PRINTH$(HE)
13030 NEXT HE
13035 GETR$:IFR$=""THEN13035
13036 IF R$<>CHR$(13) THEN 13035
13040 GOSUB11000:PRINT"                              ▲‾⎸_⊥+⎸ : ♠⎸/‾♠▀";
13045 X=CO:Y=CV
13050 RETURN
15000 REM ----- ACTUALIZACION -----
15005 GOSUB6000:GOSUB6200
15007 OZ=2:PRINTHO$:VO$="   ♠CTUALIZACION ":GOSUB 7003
15010 FOR K9=1 TO 10:PA=K9:GOSUB6800:INPUT#2,B$(K9):NEXT K9
15015 IN=CY(1):TP(1)=5
15020 FOR K=2 TO 10:TP(K)=TP(1):LC(K)=LC(1):ND(K)=ND(1):CX(K)=CX(1)
15025 IN=IN+1:CY(K)=IN:NEXT K
15027 NUC=10
15028 D$(1)="XARIACION ATRIBUTOS"
15030 GOSUB 2000
15032 J=1
15035 GOSUB 3040
15037 IF FS=9 THEN 15035
15038 IF FS=7 THEN RUN 20
15040 IF FS=6 THEN 15030
15045 FOR K9=1 TO 10:PA=K9:GOSUB6800:PRINT#2,B$(K9):NEXT K9
15050 GOSUB6400:GOSUB6600
15055 B=1:FOR K=2 TO 20STEP2:PK$(K)=B$(B):B=B+1:NEXT K
15060 REM --- ACTUALIZACION ELEMENTOS ---
15065 NF$="EL"+S$:GOSUB6000:GOSUB6200
15070 PA=1:GOSUB6800:INPUT#2,AA$:MX=VAL(AA$)
15072 OZ=3:PRINTHO$:GOSUB 7003
15075 FOR NE=2 TO MX
15076 OZ=1:PRINTHO$:GOSUB 7003
15080 PA=NE:GOSUB6800:INPUT#2,AA$
15085 B$(1)=LEFT$(AA$,10):P$=RIGHT$(AA$,10)
15087 B=3
15088 FOR JN=1 TO 10:B$(B)=MID$(P$,JN,1):B=B+2:NEXT JN
15090 IN=CY(2):B=2
15095 FOR K=2 TO 21STEP2
15100 B$(K)=PK$(B)
15105 TP(K)=4:TP(K+1)=5:LC(K)=LC(2):LC(K+1)=LC(3):ND(K)=ND(2)
15110 ND(K+1)=ND(3):CX(K)=CX(2):CX(K+1)=CX(3)
15115 IN=IN+1:CY(K)=IN:CY(K+1)=IN:B=B+2
15120 NEXT K
15125 NUC=21
15127 X=9:Y=21:GOSUB11000:PRINT"▮_7 = SIGUIENTE ELEMENTO"
15130 GOSUB 2000
15135 J=1
15140 GOSUB 3040
15142 IF FS=6 THEN 15130
15144 IF FS=9 THEN 15140
15145 IF FS=7 THEN 15190
15150 A$=""
15155 FOR K=3 TO 21STEP2:A$=A$+B$(K):NEXT K
15160 AA$=B$(1)+A$
15165 PA=NE:GOSUB6800:PRINT#2,AA$
15190 NEXT NE
```

```
15200 GOSUB6400:GOSUB6600
15250 RUN 20
20000 REM RESTORE DATA
20005 RESTORE
20045 IF LV<=0 THEN RETURN
20050 FOR I=1 TO LV:READ LV$:NEXTI
20100 RETURN
```

UA

```
12 REM ************* MAIN *************
13 PRINT"□"
15 X=7:Y=11:GOSUB11000:PRINT"■INTRODUCIR EL DISCO DATOS":Y=Y+3:X=X+3
16 GOSUB11000:PRINT"Y PULSAR UNA TECLA"
17 GET A$:IF A$<>"" THEN PRINTHO$:GOTO20
18 FOR T=1 TO 400:NEXTT:GOTO13
20 DIM B$(25),TP(25),LC(25),ND(25),CX(25),CY(25),DX(15),DY(15)
21 DIM TF(15),D$(15),H$(12),NO$(100),FO$(100),DT$(10),PU(10),PK$(21)
24 BL$=" ":PD$=".":AS$="-":NS$="#":SU$="■":GIU$="■":HO$="□":SL$="/"
25 POKE 650,128
26 PRINTHO$
27 REM *** LECTURAS DE SERVICIO ***
30 RESTORE
35 FOR I=1 TO 15:READ TF(I):NEXTI
40 FOR I=1 TO 12:READ H$(I):NEXTI
45 REM
100 PRINTHO$:OZ=1:GOSUB 7000
110 X=2:Y=10:GOSUB11000:PRINT"■NTRODUCIR LA SIGLA "
115 Y=Y+2:GOSUB11000:PRINT"■DEL TEST (2 CARACTERES)■ --■":X=26
117 S$=""
119 Y=Y+1:GOSUB11000:PRINT"■↑":Y=Y-1
120 FOR I=1 TO 2
122 GET L$:IF L$="" THEN 122
123 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
124 GOSUB11000:PRINT "■"L$:X=X+1
126 Y=Y+1:GOSUB11000:PRINT"■↑":Y=Y-1
132 S$=S$+L$
135 NEXT I
140 Y=Y+1:GOSUB11000:PRINT" ":Y=Y-1
152 NF$="AT"+S$:GOSUB 6900:IF A=62 THEN PRINTHO$:GOSUB9006:RUN 20
155 GOSUB6000:GOSUB6200
160 FOR PA=1 TO 10:GOSUB6800:INPUT#2,DT$(PA):NEXT PA
165 GOSUB6600:GOSUB6400
170 PRINTHO$:GOSUB7000:X=2:Y=10:GOSUB11000:INPUT"■LECTURA ATRIBUTOS (S/N) ";R$
172 IF R$="N" OR R$="/" THEN 280
175 IF R$<>"S" AND R$<>"♥" THEN 170
180 X=2:Y=10:GOSUB11000:PRINT"                              "
182 X=5:Y=8
184 FOR K=1 TO 10:Y=Y+1:GOSUB11000:PRINTDT$(K):NEXTK
186 X=0:Y=23:GOSUB11000:PRINT"■PULSAR UNA TECLA AL TERMINAR LA LECTURA":POKE 198
,0
188 WAIT 198,1:POKE198,0
280 REM --- LECTURA ELEMENTOS ---
285 NF$="EL"+S$
297 GOSUB6000:GOSUB6200:PA=1:GOSUB6800:INPUT#2,AA$:MX=VAL(AA$)
300 FOR PA=2 TO MX:GOSUB6800:INPUT#2,AA$
305 NO$(PA)=LEFT$(AA$,20):FO$(PA)=RIGHT$(AA$,10)
310 NEXT PA:GOSUB6400:GOSUB6600
530 REM ------ TIPO DE TEST ------
550 PRINTHO$:GOSUB7000
555 X=7:Y=10:GOSUB11000:PRINT"■ESARROLLO POR LINEA "
556 Y=Y+2:GOSUB11000:PRINT"■O POR COLUMNA (L/-)   ";:INPUT"■";R$
560 IF R$="L" OR R$="L" THEN GOSUB 15000:GOTO 570
565 IF R$="C" OR R$="-" THEN GOSUB 15500:GOTO 570
567 GOTO 550
570 REM --- CONTROL REALIZADO ---
575 PRINTHO$:OZ=1:GOSUB7000
580 X=5:Y=12:GOSUB11000:PRINT"■TRO TEST (S/N)";:INPUT "■";R$
585 IF R$="N" OR R$="/" THEN PRINTHO$:POKE 646,6:CLR:LOAD"TEST",8
590 IF R$<>"S" AND R$<>"♥" THEN 575
600 GOTO 550
2000 REM
2005 REM *** PRESENTA MASCARA ***
2015 FORI=1TODS:X=DX(I):Y=DY(I):GOSUB11000:PRINT"■"D$(I):NEXTI
2030 FOR J=1 TO NUC
2032 A$=""
2050 IF TP(J)<>5 AND TP(J)<>4 THEN 2060
2054 X=CX(J):Y=CY(J):GOSUB11000:PRINT"■";B$(J)
2056 GOTO 2130
2060 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J)=2 THEN C$=NS$:GOTO2065
2064 C$=AS$
2065 IF N1=1 THEN A$=C$:GOTO 2095
2070 FOR I=1 TO N1-N2-1:A$=A$+C$:NEXTI
2075 IF N2=0 THEN A$=A$+C$:GOTO 2095
2080 A$=A$+PD$
2090 FOR I=1 TO N2:A$=A$+"0":NEXTI
2095 X=CX(J):Y=CY(J):GOSUB 11000
2096 PRINT "■";A$;"■"
2097 B$(J)=""
2098 FOR K=1 TO N1-N2-1
2100 B$(J)=B$(J)+NS$
2105 NEXT K
2106 IF N2=0 THEN B$(J)=B$(J)+NS$:GOTO 2130
2110 B$(J)=B$(J)+PD$
2120 FORK=1TON2:B$(J)=B$(J)+NS$:NEXTK
2130 NEXT J
2150 RETURN
3000 REM
3010 REM ** DATA ENTRY **
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3065 IF FS=9 THEN RETURN
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 REM CARACTER
3105 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 FS=0:J=NUC:GOTO 3040
3150 IF J<NUC THEN J=J+1:GOTO 3040
3160 FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB11000:GOTO 3060
3200 FS=0:GOTO 3040
3210 REM
3220 IF FS>8 THEN FS=0:GOTO 3060
3230 IF FS=8 THEN FS=0:GOTO 3100
3235 RETURN
3240 REM
3250 REM  ** LECTURA DE 1 CAMPO **
3280 FS=0
3290 IF TP(NC)=4 THEN RETURN
3300 IF TP(NC)=2 THEN GOSUB 9000:RETURN
3310 IF MID$(B$(NC),CC,1)=PD$ THENX=X+S:CC=CC+S
3320 GOSUB 11000
3330 GET A$:IF A$<>"" THEN 3338
3332 GOSUB11000:PRINT"■";:GOSUB11000:PRINT"■";MID$(B$(NC),CC,1);
3336 GOTO 3330
3338 IFASC(A$)=190RASC(A$)=200RASC(A$)=1470RASC(A$)=148 ORASC(A$)=44  THEN 3330
3340 PRINT"■■";A$;
3350 CA=ASC(A$)
3360 FOR I=1 TO 15
3370 IF TF(I)=CA THEN FS=I:I=15:GOSUB3630:RETURN
3375 IF FS=9 THEN RETURN
3380 NEXTI
3390 IF TP(NC)<>1 THEN 3450
3400 IF CA=45 OR CA=46 OR(CA>=48ANDCA<=57)THEN3420
3410 GOTO 3330
3420 IF CA<>46 THEN3450
3430 GOSUB 4270
3440 IF SWP=1 THEN 3410
3450 S=1:GOSUB 3520
3460 IF CC=LC(NC) THEN RETURN
3470 CC=CC+1:X=X+1:GOTO 3310
3520 REM
3521 REM ** TRASFERENCIA AL BUFFER **
3540 REM
3570 IF CC=1 THEN SN$="":GOTO 3590
3580 SN$=LEFT$(B$(NC),CC-1)
3590 IF CC=LC(NC) THEN DE$="":GOTO3610
3600 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3610 B$(NC)=SN$+A$+DE$
3620 RETURN
```

```
3625 REM
3630 REM ** COMANDOS **
3640 REM
3680 IF FS>5 THEN 3710
3700 RETURN
3710 IF FS=9 THEN GOSUB 13000:RETURN
3720 IF FS=8 THEN GOSUB 4080:RETURN
3725 IF FS<8 THEN RETURN
3730 ON FS-9 GOSUB 3740,3910
3735 RETURN
3740 REM
3745 REM ** INSERCION **
3750 REM
3770 IF CC=LC(NC)THEN RETURN
3780 IF CC=1 THEN SN$="":GOTO3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DE$=RIGHT$(B$(NC),N)
3820 C$=BL$
3830 A$=SN$+C$+DE$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 REM
3890 PRINT B$(NC):S=-1
3900 IF RIGHT$(B$(NC),1)=PD$ THEN 3780
3905 RETURN
3909 REM
3910 REM ***** BORRADO *****
3920 REM
3930 IF TP(NC)<>1 THEN 3950
3940 IF MID$(B$(NC),2,1)=PD$ AND CC=1 THEN RETURN
3950 IF CC<1 THEN RETURN
3960 IF CC=LC(NC) THEN DE$="":GOTO 3980
3970 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3980 SN$=LEFT$(B$(NC),CC-1)
3990 IF TP(NC)=1 OR TP(NC)=2 THEN C$=NS$:GOTO 4010
4000 C$=AS$
4010 B$(NC)=SN$+DE$+C$
4020 XX=X:YY=Y
4030 X=CX(NC):Y=CY(NC):GOSUB 11000
4040 X=XX:Y=YY
4050 PRINT B$(NC):S=1:RETURN
4080 REM ******* ALINEADOS *******
4084 IF LC(NC)=1 THEN RETURN
4085 N1=LC(NC)-CC+1
4087 IF CC=1 THEN B$(NC)="":GOTO4100
4090 B$(NC)=LEFT$(B$(NC),CC-1)
4100 IF TP(NC)=1 OR TP(NC)=2 THEN 4150
4110 FOR I=1 TO N1
4120 B$(NC)=B$(NC)+BL$
4130 NEXT I
4140 GOTO  4190
4150 FOR I=1 TO N1
4160 B$(NC)=CHR$(32)+B$(NC)
4170 NEXT I
4180 IF RIGHT$(B$(NC),1)=PD$ THEN B$(NC)=BL$+LEFT$(B$(NC),LC(NC)-1)
4190 X=CX(NC):Y=CY(NC):GOSUB 11000
4210 PRINT B$(NC):RETURN
4270 REM *** CTRL PUNTO ***
4290 SWP=0
4300 IF CC=1 OR CC=LC(NC) THEN SWP=1:RETURN
4310 FOR I=1 TO LC(NC)
4320 IF MID$(B$(NC),I,1)=BL$ THEN I=LC(NC):SWP=1:RETURN
4330 NEXT I:RETURN
4340 REM
6000 REM
6001 REM *** GESTION DISCO ***
6010 REM -----------------------
6011 REM APERTURA CANAL COMANDOS
6012 REM -----------------------
6030 OPEN1,8,15
6035 RETURN
6200 REM ----------------
6204 REM APERTURA FICHERO
6206 REM ----------------
6210 OPEN2,8,3,NF$
6215 REM
6250 RETURN
6400 REM ---------------------
6402 REM CIERRE CANAL COMANDOS
6403 REM ---------------------
6410 CLOSE1
6415 RETURN
6490 REM
6600 REM --------------
6602 REM CIERRA FICHERO
6605 REM --------------
6610 CLOSE2
6620 RETURN
6900 REM ------------------
6901 REM APUNTA AL REGISTRO
6902 REM ------------------
6905 R1=PA:R2=0
6910 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6915 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6920 RETURN
6925 REM
6900 REM CONTROLES SOBRE LOS FICHEROS
6902 GOSUB6000:GOSUB6200
6904 INPUT#1,A,B$,C,D
6906 GOSUB6400:GOSUB6600
6910 RETURN
7000 REM
7001 REM *** CARGA LOS DATA ***
7002 VO$="    ESARROLLO - ATRIBUTOS"
7003 L=LEN(VO$)-3:L$=RIGHT$(VO$,L)
7004 U$=" TEST "
7005 P=(30-L)/2:P$="":FORK=1TOP:P$=P$+" ":NEXT K
7006 PRINT"♥";:PRINTCHR$(14)
7008 PRINT"           .I.♦.   DICIONES ORUM. ♦.♠."
7011 PRINT"      ";U$;"                    "
7012 PRINT"                ";P$;
7015 PRINT"  "L$;"  ";P$
7020 PRINT"                              "
7025 IF OZ=1 THEN RETURN
7030 X=1:Y=22:GOSUB11000:PRINT"─────────────────────────────"
7033 X=25:Y=23:GOSUB11000:PRINT"  ─┤ ┴+ │ : ♠│/♠"
7034 REM
7040 ON OZ GOTO 7110,7042,7044
7042 LV=27:GOSUB20000:NUC=3:DS=2:GOTO7090
7044 LV=48:GOSUB20000:NUC=3:DS=2:GOTO7090
7090 FOR I=1 TO DS:READ D$(I),DX(I),DY(I):NEXTI
7095 FOR I=1 TO NUC:READ TP(I),LC(I),ND(I),CX(I),CY(I)
7097 REM
7100 NEXT I
7110 RETURN
9000 REM *** ERRORES FICHEROS ***
9002 IF A=100 THEN 9008
9005 X=5:Y=12:GOSUB11000:PRINT"  ┌─┌─ , ISCO FUERA DE LINEA":GOTO 9010
9006 X=5:Y=12:GOSUB11000:PRINT"  ┌─┌─ , EL FICHERO DATOS NO "
9007 Y=Y+2:GOSUB11000:PRINT"ESTA EN ESTE DISCO":GOTO 9010
9008 X=5:Y=12:GOSUB11000:PRINT"  ┌─┌─ , ESPACIO DATOS AGOTADO"
9010 X=6:Y=21:GOSUB11000:PRINT" PULSAR UNA TECLA PARA VOLVER"
9015 POKE198,0:WAIT198,1
9016 IF A=70 THEN POKE 198,0:RUN 20
9020 POKE198,0:RETURN
9150 REM ***** INDICA ERROR *********
9152 X=10:Y=21:GOSUB11000:PRINT"  ─┌─ ─ ╲/┘┌─ ╱─╲┌/"
9154 FOR T=1 TO3000:NEXTT
9156 X=10:Y=21:GOSUB11000:PRINT"                          "
9160 RETURN
10000 REM ************ DATA ***********
10001 DATA 145,17,29,157,140,133,136,13,25,134,135,0,0,0,0
10005 REM
10007 DATA"    < ─┐ ╱─╱ >        ♠─♠ ┬/│╲/♠─"
10010 DATA"    ─♦─ ─╲│ ||  :  ─♥┐♠♦♠ ♠ ─┬─┬│♠"
10012 DATA"   ─♦─ └─┘    :   ─♥┐♠♦♠ ♠ ╲♦♦ ∧─┬♠"
10014 DATA"    ─♦─ ╱┐     :   ─♥┐♠♦♠ ♠──╲│♠      "
10016 DATA"    ─♦─ ┬O/   :   ─♥┐♠♦♠ ♠│♠┬   "
10018 DATA"        ╯─1╯  :  ♠/┘♠ ╲/┘┌─ ╱─╲┌/"
10020 DATA"        ╯─3╯  :  ╲/♥─│♠ ─♠─♠─│─  "
10022 DATA"        ╯─5╯  :  │┌─♠ ─♠─♠─│─  "
10024 DATA"        ╯─7╯  :  X/┐X ♠L ╲┐/╱    "
10026 DATA"        ╯─8╯  :  ╲/┘┌─╱─ L♠ ─♥┐/♥│♠"
10028 DATA"    ─│ ┴+│     :  ┐─ L♠ ♠│/♠     "
10030 DATA"  ─┐ ╱─╱       :  ♠L╲/♠ │ ┬/X♠L╲─♠"
10040 REM ********* OZ=2 *********
```

```
10042 DATA"ATRIBUTO :",4,6
10043 DATA"LEMENTOS : ",4,9
10044 DATA4,30,0,8,7
10045 DATA4,20,0,4,9
10046 DATA1,1,0,26,9
10050 REM ********* OZ=3 *********
10052 DATA"LEMENTO : ",4,7
10053 DATA"ATRIBUTO",4,9
10055 DATA4,20,0,18,7
10056 DATA4,30,0,2,9
10057 DATA1,1,0,36,9
11000 REM
11001 REM **** POSICIONADO EN X,Y ****
11005 POKE 211,X
11010 POKE 214,Y
11015 SYS58732
11020 RETURN
13000 REM *** AYUDA ***
13005 CO=X:CV=Y
13010 X=1:Y=23
13015 GOSUB11000:PRINTH$(1);"      ";
13017 FOR HE=2 TO 12
13020 GETR$:IFR$=""THEN13020
13021 IF R$<>CHR$(13) THEN 13020
13022 GOSUB11000:PRINTH$(HE)
13030 NEXT HE
13035 GETR$:IFR$=""THEN13035
13036 IF R$<>CHR$(13) THEN 13035
13040 GOSUB11000:PRINT"                              ";
13045 X=CO:Y=CV
13050 RETURN
15000 REM --- DESARROLLO POR LINEA ---
15005 GOSUB21000:REM GENERA N9
15010 IF LEFT$(NO$(N9),1)=CHR$(32) OR LEFT$(NO$(N9),1)=CHR$(35) THEN 15005
15015 B$(1)=NO$(N9):OZ=3:PRINTHO$:GOSUB7000
15020 B=2:FOR K9=1 TO 10:PK$(B)=DT$(K9):B=B+2:NEXT K9
15022 IN=CY(2):B=2
15025 FOR K=2 TO 21STEP2
15027 B$(K)=PK$(B)
15030 TP(K)=TP(2):TP(K+1)=3:LC(K)=LC(2):LC(K+1)=1:ND(K)=ND(2):ND(K+1)=0
15032 CX(K)=CX(2):CX(K+1)=36:IN=IN+1:CY(K)=IN:CY(K+1)=IN:B=B+2
15034 NEXT K
15036 NUC=21
15037 X=7:Y=5:GOSUB11000:PRINT"EMPLEAR '♥' PARA SI Y '/' PARA NO"
15040 GOSUB 2000
15042 J=1
15045 GOSUB 3000
15050 IF FS=6 OR FS=7 THEN 15040
15052 IF FS=9 THEN 15045
15055 REM --- RESPUESTAS ---
15056 ER=0:R$=""
15057 FOR K9=3 TO 21STEP2
15060 IF B$(K9)="S" OR B$(K9)="♥" THEN B$(K9)="1":GOTO 15080
15065 IF B$(K9)="N" OR B$(K9)="/" THEN B$(K9)="0":GOTO 15080
15070 ER=ER+1
15080 R$=R$+B$(K9):NEXT K9
15085 IF ER<>0 THENGOSUB9150:GOTO 15045
15087 ER=10
15090 FOR K9=1 TO 10
15095 C1$=MID$(FO$(N9),K9,1)
15100 C2$=MID$(R$,K9,1)
15105 IF C1$=C2$ THEN ER=ER-1
15110 NEXT K9
15115 IF ER<>0 THEN GOSUB16000:GOTO 15015
15120 GOSUB16100
15125 RETURN
15500 REM --- DESARROLLO POR COLUMNA ---
15505 NN=10:N9=INT(RND(0)*NN+1)
15510 B$(1)=DT$(N9):B=2:RO$=""
15512 FOR K9=1TO10:PU(K9)=0:NEXTK9
15515 FOR K=1 TO 10
15520 KK=MX:N8=INT(RND(0)*KK+1):IF N8=1 THEN 15520
15521 IF MX<=10 THEN 15525
15522 IF PU(K)<>0 THEN 15520
15525 IF LEFT$(NO$(N8),1)=CHR$(32) OR LEFT$(NO$(N8),1)=CHR$(35) THEN 15520
15530 B$(B)=NO$(N8):PU(K)=K
15535 RO$=RO$+MID$(FO$(N8),N9,1)
15540 B=B+2:NEXT K
15542 OZ=2:PRINTHO$:GOSUB7000
15543 IN=CY(2)
15545 FOR K=2 TO 21STEP2
15550 TP(K)=4:TP(K+1)=3:LC(K)=20:LC(K+1)=1:ND(K)=0:ND(K+1)=0
15555 CX(K)=CX(2):CX(K+1)=CX(3):IN=IN+1:CY(K)=IN:CY(K+1)=IN:NEXT K
15560 NUC=21
15565 X=7:Y=5:GOSUB11000:PRINT"EMPLEAR '♥' PARA SI Y '/' PARA NO"
15570 GOSUB2000
15572 J=1
15575 GOSUB3000
15580 IF FS=6 OR FS=7 THEN 15570
15582 IF FS=9 THEN 15575
15585 REM --- RESPUESTAS ---
15590 ER=0:R$=""
15595 FOR K9=3 TO 21STEP2
15600 IF B$(K9)="S" OR B$(K9)="♥" THEN B$(K9)="1":GOTO 15615
15605 IF B$(K9)="N" OR B$(K9)="/" THEN B$(K9)="0":GOTO 15615
15610 ER=ER+1
15615 R$=R$+B$(K9):NEXT K9
15620 IF ER<>0 THENGOSUB9150:GOTO 15570
15625 ER=10
15630 FOR K9=1 TO 10
15635 C1$=MID$(RO$,K9,1)
15640 C2$=MID$(R$,K9,1)
15645 IF C1$=C2$ THEN ER=ER-1
15650 NEXT K9
15655 IF ER<>0 THEN GOSUB16000:GOTO 15542
15660 GOSUB16100
15665 RETURN
16000 REM --- ERROR ---
16005 PRINTHO$:X=17:Y=12:GOSUB11000:PRINT"     "
16010 X=12:Y=22:GOSUB11000:PRINT"PULSAR UNA TECLA":POKE198,0
16015 WAIT198,1:POKE198,0:RETURN
16100 REM --- CORRECTO ---
16105 PRINTHO$:X=13:Y=12:GOSUB11000:PRINT"        "
16110 X=12:Y=22:GOSUB11000:PRINT"PULSAR UNA TECLA":POKE198,0
16115 WAIT198,1:POKE198,0:RETURN
20000 REM RESTORE DATA
20005 RESTORE
20045 IF LV<=0 THEN RETURN
20050 FOR I=1 TO LV:READ LV$:NEXTI
20100 RETURN
21000 REM --- RANDOM ENTRE 1 Y MX-1 ---
21005 K9=MX
21010 N9=INT(RND(0)*K9+1)
21012 IF N9=1 THEN 21010
21015 RETURN
```

GS

```
10 REM ****** GENERACION SIMBOLOS ******
13 HO$="J":PRINTHO$:POKE53281,6:POKE53280,14:POKE650,128
14 X=7:Y=12:GOSUB1054:PRINT"INTRODUCIR EL DISCO DATOS ":Y=Y+2:X=10
15 GOSUB1054:PRINT"Y PULSAR UNA TECLA":POKE198,0:WAIT198,1:POKE198,0
16 PRINTHO$:PRINTCHR$(14)
17 DIM SP$(62,5),TA(7),LZ(5)
18 J=1:FOR K=195 TO 199:LZ(J)=K:J=J+1:NEXTK
19 TA(1)=145:TA(2)=17:TA(3)=157:TA(4)=29:TA(5)=69:TA(6)=65:TA(7)=82
20 NF$="SIMBOLOS":GOSUB1032
21 IF A=0 THEN 30
22 IF A=74 THEN GOSUB 1045:RUN 16
24 GOSUB1003:OPEN2,8,3,"0:SIMBOLOS,L,"+CHR$(65)
25 AA$="01":FOR K=1 TO 61:AA$=AA$+"#":NEXT K
26 PA=1:GOSUB1025:PRINT#2,AA$
27 GOSUB 1014:GOSUB 1020
30 GOSUB1003:GOSUB1008:PA=1:GOSUB1025:INPUT#2,AA$:MX=VAL(AA$)
35 NRM=50:GOSUB 1039
37 IN=2+16*(MX-1)
40 FOR NS=1 TO 5
41 GOSUB 1059:REM PREPARA SPRITE #NS
42 X=1:Y=24:GOSUB1054:PRINT"ONFIRMA EL SPRITE (S/N) ?";:GETY$:IF Y$="" THEN42
43 IF Y$="/" OR Y$="N" THEN 41
44 IF Y$<>"♥" AND Y$<>"S" THEN 42
45 NEXT NS
50 GOSUB900:REM PRESENTACION TOTAL
55 GOSUB800
100 X=1:Y=20:GOSUB1054:PRINT"ESPUESTA PREVISTA (4-6) ?":POKE 198,0
105 GET R$:IF R$="" THEN 105
106 R=VAL(R$):IF R<4 OR R>6 THEN 105
```

```
108 X=28:GOSUB1054:PRINT"■";R
109 X=1:Y=22:GOSUB1054:PRINT"■EGURO (S/N) ";:INPUT S$
110 IF S$="N" OR S$="/" THEN GOSUB350:GOTO100
111 IF S$<>"♥" AND S$<>"S" THEN 109
115 RI$="0"+RIGHT$(STR$(R),1)
116 FOR K=1 TO 61:RI$=RI$+"#":NEXT K
120 REM --- ESCRIBE REGISTROS ---
122 PR=IN
125 FOR J9=1 TO 5
130 AA$=""
135 FOR K9=0 TO 20:AA$=AA$+SP$(K9,J9):NEXT K9
140 PA=PR:GOSUB 1025:PRINT#2,AA$
142 AA$=""
145 FOR K9=21 TO 41:AA$=AA$+SP$(K9,J9):NEXT K9
150 PA=PA+1:GOSUB 1025:PRINT#2,AA$
152 AA$=""
155 FOR K9=42 TO 62:AA$=AA$+SP$(K9,J9):NEXT K9
160 PA=PA+1:GOSUB 1025:PRINT#2,AA$
170 PR=PR+3
175 NEXT J9
185 PA=PA+1:GOSUB1025:PRINT#2,RI$
190 CLOSE1:CLOSE2
200 REM
202 POKESP+21,PEEK(SP+21)AND254
203 POKESP+21,PEEK(SP+21)AND253
204 POKESP+21,PEEK(SP+21)AND251
205 POKESP+21,PEEK(SP+21)AND247
206 POKESP+21,PEEK(SP+21)AND239
208 MX=MX+1:PRINTHO$
210 X=2:Y=12:GOSUB1054:PRINT"■TRA INTRODUCCION (S/N) ";:INPUT"■";F$
212 IF F$="N" OR F$="/" THEN 240
214 IF F$<>"S" AND F$<>"♥" THEN 210
240 GOSUB 1003:GOSUB 1008
250 IF MX<10 THEN MX$="0"+RIGHT$(STR$(MX),1):GOTO 260
255 MX$=RIGHT$(STR$(MX),2)
260 FOR K=1 TO 61:MX$=MX$+"#":NEXT K
265 PA=1:GOSUB1025:PRINT#2,MX$
270 GOSUB 1014:GOSUB 1020
274 IF F$="S" OR F$="♥" THEN RUN
275 PRINTHO$:POKE646,6:CLR:LOAD"TEST",8
350 REM
355 X=28:Y=20:GOSUB1054:PRINT"       "
360 X=1:Y=22:GOSUB1054:PRINT"                                        ":RETURN
800 REM ** ACTIVA REGISTROS SPRITES **
802 SP=53248
804 POKE SP+21,31
806 J=1:FOR K=2040 TO 2044:POKE K,LZ(J):J=J+1:NEXTK
808 FOR K=39 TO 43:POKE SP+K,1:NEXT K
810 POKE SP+23,31:POKE SP+29,31
812 POKE SP,65:POKE SP+1,80
814 POKE SP+2,140:POKE SP+3,80
816 POKE SP+4,65:POKE SP+5,145
818 POKE SP+6,140:POKE SP+7,145
820 POKE SP+8,220:POKE SP+9,145
822 X=3:Y=9:GOSUB1054:PRINT"■1"
823 X=13:Y=9:GOSUB1054:PRINT"2"
824 X=3:Y=17:GOSUB1054:PRINT"4"
825 X=13:Y=17:GOSUB1054:PRINT"5"
826 X=23:Y=17:GOSUB1054:PRINT"6"
850 RETURN
900 REM ** CABECERA **
905 PRINTHO$:X=1:Y=0:GOSUB1054:PRINT"■NTRODUCCION TEST CON SIMBOLOS N.";MX:RETU
RN
1003 REM APERTURA CANAL COMANDOS
1005 OPEN1,8,15
1006 RETURN
1008 REM APERTURA FICHERO
1010 OPEN2,8,3,NF$:RETURN
1014 REM CIERRE CANAL COMANDOS
1016 CLOSE1
1017 RETURN
1020 REM CIERRA FICHERO
1022 CLOSE2
1023 RETURN
1025 REM APUNTA AL REGISTRO
1027 R1=PA:R2=0
1028 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
1029 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
1030 RETURN
1032 REM CONTROLES SOBRE LOS FICHEROS
1033 GOSUB1003:GOSUB1008
1034 INPUT#1,A,B$,C,D
1035 GOSUB1014:GOSUB1020
1036 RETURN
1039 REM CONTROL FIN FICHEROS
1040 IF MX<NRM THEN RETURN
1041 PRINTHO$
1042 X=10:Y=10:GOSUB1054:PRINT"EL FICHERO DATOS ESTA AGOTADO"
1043 X=2:Y=20:GOSUB1054:PRINT"PULSE UNA TECLA PARA VOLVER AL MENU"
1044 POKE198,0:WAIT198,1:RUN
1045 REM *** ERRORES FICHEROS ***
1047 X=5:Y=12:GOSUB1054:PRINT"■┌─┌─ , ¯ISCO FUERA DE LINEA":GOTO 1050
1048 X=5:Y=12:GOSUB1054:PRINT"■┌─┌─ , EL FICHERO DATOS"
1049 Y=Y+2:GOSUB1054:PRINT"YA ESTA CREADO EN ESTE DISCO"
1050 X=8:Y=21:GOSUB1054:PRINT"■PULSE UNA TECLA PARA CONTINUAR"
1051 POKE198,0:WAIT198,1
1052 POKE198,0:RETURN
1054 REM **** POSICIONADO EN X,Y ****
1055 POKE 211,X
1056 POKE 214,Y
1057 SYS58732
1058 RETURN
1059 REM ** CREACION FIGURA **
1060 PRINTHO$;CHR$(14):POKE53280,6
1061 FF=NS:IF NS>2 THEN FF=NS+1
1062 P1=LZ(NS)*64
1063 FOR I=0 TO 62:POKEP1+I,0:NEXTI
1064 SP=53248:POKESP+21,1:POKE2040,LZ(NS):POKESP+39,1:POKESP+23,1:POKESP+29,1
1065 POKESP,255:POKESP+1,60
1066 X=1:Y=0
1067 FOR J=1 TO 24
1068 X=X+1:GOSUB1054
1069 J$=STR$(J):IF LEN(J$)=3 THEN J$=RIGHT$(J$,2)
1070 PRINT"■";LEFT$(J$,1)
1071 Y=Y+1:GOSUB1054:PRINT"■";RIGHT$(J$,1)
1072 Y=Y-1
1073 NEXT J
1074 X=1:Y=0:GOSUB1054:PRINT"■ ":Y=Y+1:GOSUB1054:PRINT"■ "
1075 X=26:Y=0:GOSUB1054:PRINT"■ ":Y=Y+1:GOSUB1054:PRINT"■ "
1076 X=1:Y=1
1077 FOR J=65 TO J+20
1078 Y=Y+1:GOSUB1054:PRINT"■";CHR$(J);SPC(24)"■ "
1079 NEXT J
1080 PRINT" ■                          "
1081 X=1:Y=24:GOSUB1054:PRINT"■IMBOLO N.";FF;
1082 X=28:Y=0:GOSUB1054:PRINT"■ ┌────────┐"
1083 FOR K1=1 TO 7
1084 Y=Y+1:GOSUB1054:PRINT"|        |"
1085 NEXT K1
1086 GOSUB1054:PRINT"└────────┘"
1087 K=1066
1088 FOR J=1 TO 21
1089 K=K+40
1090 FOR I=K TO K+23:POKEI,160:POKEI+54272,14:NEXTI
1091 NEXT J
1092 REM
1093 REM
1094 Y=Y+1:GOSUB1054:PRINT"■■ :ENCIENDE";
1095 Y=Y+1:GOSUB1054:PRINT"■■ :APAGA ";
1096 Y=Y+1:GOSUB1054:PRINT"■■ :INTRODUCE ";
1097 X0=2:Y0=2:XF=25:YF=22:X=X0:Y=Y0:S1=1:S2=1:CC=14
1098 GETA$:IFA$<>""THEN1104
1099 GOSUB1054:PRINT"■■ ";
1100 FOR T=1 TO 30:NEXTT
1101 GOSUB1054:POKE646,CC:PRINT"■ ";
1102 FOR T=1 TO 30:NEXTT
1103 GOTO 1098
1104 GOSUB 1107:IF FA=7 THEN RETURN
1105 GOTO 1098
1106 REM -------------------------
1107 REM CONTROL TECLA
1108 FA=0:K9=0
1109 CA=ASC(A$)
1110 FOR I=1 TO 7
1111 IF CA=TA(I) THEN FA=I:I=7
1112 NEXT I
1113 IF FA=0 THEN RETURN
1114 ON FA GOSUB 1116,1121,1126,1131,1136,1144,1152
```

```
1115 RETURN
1116 Y=Y-1:IF Y<Y0 THEN Y=Y+1:RETURN
1117 S2=S2-1
1118 GOSUB1136:IF (PEEK(P1+BT)AND2↑BI)>0 THEN CC=0:GOTO 1120
1119 CC=14
1120 RETURN
1121 Y=Y+1:IF Y>YF THEN Y=Y-1:RETURN
1122 S2=S2+1
1123 GOSUB1136:IF (PEEK(P1+BT)AND2↑BI)>0 THEN CC=0:GOTO 1125
1124 CC=14
1125 RETURN
1126 X=X-1:IF X<X0 THEN X=X+1:RETURN
1127 S1=S1-1
1128 GOSUB1136:IF (PEEK(P1+BT)AND2↑BI)>0 THEN CC=0:GOTO 1130
1129 CC=14
1130 RETURN
1131 X=X+1:IF X>XF THEN X=X-1:RETURN
1132 S1=S1+1
1133 GOSUB1136:IF (PEEK(P1+BT)AND2↑BI)>0 THEN CC=0:GOTO 1135
1134 CC=14
1135 RETURN
1136 REM ENCIENDE EN EL SPRITE
1137 BT=(S2-1)*3+INT((S1-1)/8)
1138 BI=8-S1+INT((S1-1)/8)*8
1139 IF FA<>5 THEN RETURN
1140 POKE P1+BT,PEEK(P1+BT) OR 2↑BI
1141 IF (PEEK(P1+BT)AND2↑BI)>0 THEN CC=0
1142 IF X=XF+1 THEN X=2:Y=Y+1:RETURN
1143 RETURN
1144 REM APAGA EN EL SPRITE
1145 BT=(S2-1)*3+INT((S1-1)/8)
1146 BI=8-S1+INT((S1-1)/8)*8
1147 IF FA<>6 THEN RETURN
1148 POKE P1+BT,PEEK(P1+BT) AND (255-2↑BI)
1149 IF (PEEK(P1+BT)OR2↑BI)>0 THEN CC=14
1150 IF X=XF+1 THEN X=2:Y=Y+1:RETURN
1151 RETURN
1152 REM * TRASFERENCIA A CADENAS *
1153 FOR I=0 TO 62
1154 B1=PEEK(P1+I):SP$(I,NS)=STR$(B1)
1155 IF B1<10 THEN SP$(I,NS)="00"+RIGHT$(SP$(I,NS),1):GOTO 1158
1156 IF B1<100 THEN SP$(I,NS)="0"+RIGHT$(SP$(I,NS),2):GOTO1158
1157 SP$(I,NS)=RIGHT$(SP$(I,NS),3)
1158 NEXT I
1159 RETURN
```

US

```
10 REM ****** DESARROLLO SIMBOLOS ******
13 HO$="□":PRINTHO$:POKE53281,6:POKE53280,14:POKE650,128
14 X=7:Y=12:GOSUB1054:PRINT"■INTRODUCIR EL DISCO DATOS ":Y=Y+2:X=10
15 GOSUB1054:PRINT"Y PULSAR UNA TECLA":POKE198,0:WAIT198,1:POKE198,0
16 HO$="□":PRINTHO$:PRINTCHR$(14)
17 DIM SP$(62,5),LZ(5)
18 J=1:FOR K=195 TO 199:LZ(J)=K:J=J+1:NEXTK
20 NF$="SIMBOLOS":GOSUB1032
21 IF A=0 THEN 30
22 IF A=74 THEN 1045
24 IF A=62 THEN 1048
30 GOSUB1003:GOSUB1008:PA=1:GOSUB1025:INPUT#2,AA$:MX=VAL(AA$)
35 TE=1
37 IN=2+16*(TE-1)
120 REM --- LEE LOS REGISTROS ---
122 PR=IN
125 FOR J9=1 TO 5
140 PA=PR:GOSUB 1025:INPUT#2,AA$
142 GOSUB 1059
150 PA=PA+1:GOSUB 1025:INPUT#2,AA$
152 GOSUB 1156
160 PA=PA+1:GOSUB 1025:INPUT#2,AA$
162 GOSUB 1160
170 PR=PR+3
175 NEXT J9
185 PA=PA+1:GOSUB1025:INPUT#2,RI$
190 RI=VAL(RI$)
200 GOSUB 1200
205 GOSUB 800
210 X=1:Y=20:GOSUB1054:PRINT"■UE  RESPUESTA  (4-6)  ?":POKE 198,0
212 GET R$:IF R$="" THEN 212
214 R=VAL(R$):IF R<4 OR R>6 THEN 212
216 X=28:GOSUB1054:PRINT"■";R
218 X=1:Y=22:GOSUB1054:PRINT"■EGURO (S/N) ";:INPUT S$
220 IF S$="N" OR S$="/" THEN GOSUB550:GOTO 210
222 IF S$<>"S" AND S$<>"♥" THEN 218
230 IF R=RI THEN GOSUB 600:GOTO 240
235 GOSUB 670:GOTO210
240 PRINTHO$
241 POKE SP+21,PEEK(SP+21)AND254
242 POKE SP+21,PEEK(SP+21)AND253
243 POKE SP+21,PEEK(SP+21)AND251
244 POKE SP+21,PEEK(SP+21)AND247
245 POKE SP+21,PEEK(SP+21)AND239
246 POKE SP+K1,K3:POKE SP+K2,K4:POKE SP+CC,1
250 X=2:Y=12:GOSUB1054:PRINT"■TRO TEST (S/N) ■";:INPUT F$
252 IF F$="N" OR F$="/" THEN 300
254 IF F$<>"S" AND F$<>"♥" THEN 250
256 TE=TE+1:IF TE<MX THEN 37
260 X=10:Y=Y+5:GOSUB1054:PRINT"■ICHERO DATOS TERMINADO"
265 X=5:Y=22:GOSUB1054:PRINT"■PULSAR UNA TECLA PARA CONTINUAR"
270 POKE198,0:WAIT198,1
300 GOSUB1014:GOSUB1020:PRINTHO$:POKE646,6:CLR:LOAD"TEST",8
550 REM
555 X=28:Y=20:GOSUB1054:PRINT"       "
560 X=1:Y=22:GOSUB1054:PRINT"                                        ":RETURN
600 X=0:Y=20:GOSUB1054:PRINT"               ■TEST  SUPERADO■          ";
605 IF R=4 THEN K1=4:K2=5:CC=41:K3=65:K4=145:GOTO 610
606 IF R=5 THEN K1=6:K2=7:CC=42:K3=140:K4=145:GOTO 610
607 IF R=6 THEN K1=8:K2=9:K3=220:K4=145:CC=43
610 X=28:Y=5:GOSUB1054:PRINT"  "
615 POKE SP+K1,220:POKE SP+K2,80:POKE SP+CC,10
655 X=1:Y=22:GOSUB1054:PRINT"■    PULSAR UNA TECLA PARA CONTINUAR    ";
660 GET H$:IF H$=""ORH$=CHR$(13) THEN 660
665 X=0:Y=22:GOSUB1054:PRINT"                                        ";:RETURN
670 X=0:Y=20:GOSUB1054:PRINT"                 ■                      ";:GOTO655
800 REM ** ACTIVACION REGISTROS SPRITES Y POSICIONES EN PANTALLA **
801 PRINTHO$:GOSUB900
802 SP=53248
804 POKE SP+21,31
806 J=1:FOR K=2040 TO 2044:POKE K,LZ(J):J=J+1:NEXTK
808 FOR K=39 TO 43:POKE SP+K,1:NEXT K
810 POKE SP+23,31:POKE SP+29,31
812 POKE SP,65:POKE SP+1,80
814 POKE SP+2,140:POKE SP+3,80
816 POKE SP+4,65:POKE SP+5,145
818 POKE SP+6,140:POKE SP+7,145
```

```
820 POKE SP+8,220:POKE SP+9,145
822 X=3:Y=9:GOSUB1054:PRINT"■1"
823 X=13:Y=9:GOSUB1054:PRINT"2"
824 X=23:Y=9:GOSUB1054:PRINT"3":X=X+5:Y=Y-4:GOSUB1054:PRINT"■?"
825 X=3:Y=17:GOSUB1054:PRINT"4"
826 X=13:Y=17:GOSUB1054:PRINT"5"
827 X=23:Y=17:GOSUB1054:PRINT"6"
850 RETURN
900 REM ** CABECERA **
905 PRINTHO$:X=1:Y=0:GOSUB1054:PRINT"■ESARROLLO TEST CON SIMBOLOS N.";TE
910 RETURN
1003 REM APERTURA CANAL COMANDOS
1005 OPEN1,8,15
1006 RETURN
1008 REM APERTURA FICHERO
1010 OPEN2,8,3,NF$:RETURN
1014 REM CIERRE CANAL COMANDOS
1016 CLOSE1
1017 RETURN
1020 REM CIERRA FICHERO
1022 CLOSE2
1023 RETURN
1025 REM APUNTA AL REGISTRO
1027 R1=PA:R2=0
1028 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
1029 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
1030 RETURN
1032 REM CONTROLES SOBRE LOS FICHEROS
1033 GOSUB1003:GOSUB1008
1034 INPUT#1,A,B$,C,D
1035 GOSUB1014:GOSUB1020
1036 RETURN
1045 REM *** ERRORES FICHEROS ***
1047 X=5:Y=12:GOSUB1054:PRINT"■┌──┌─ , "ISCO FUERA DE LINEA":GOTO 1050
1048 X=5:Y=12:GOSUB1054:PRINT"■┌──┌─ , EL FICHERO NO ESTA"
1049 Y=Y+2:GOSUB1054:PRINT"EN ESTE DISCO"
1050 X=6:Y=21:GOSUB1054:PRINT"■PULSE UNA TECLA PARA VOLVER"
1051 POKE198,0:WAIT198,1
1052 POKE198,0:RUN 16
1054 REM **** POSICIONADO EN X,Y ****
1055 POKE 211,X
1056 POKE 214,Y
1057 SYS58732
1058 RETURN
1059 REM ** LECTURA FIGURAS **
1153 B=1
1154 FOR I=0 TO 20:SP$(I,J9)=MID$(AA$,B,3):B=B+3:NEXT I
1155 RETURN
1156 B=1
1157 FOR I=21 TO 41:SP$(I,J9)=MID$(AA$,B,3):B=B+3:NEXT I
1159 RETURN
1160 B=1
1161 FOR I=42 TO 62:SP$(I,J9)=MID$(AA$,B,3):B=B+3:NEXT I
1162 RETURN
1200 REM SPRITE EN MEMORIA
1205 FOR J=1 TO 5
1210 BL=LZ(J)*64
1212 FOR K9=0 TO 62
1214 POKE BL+K9,VAL(SP$(K9,J))
1216 NEXT K9
1218 NEXT J
1220 RETURN
```

RELACION DE VARIABLES - COMMODORE 64

GP

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud del campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en el vídeo del campo de introducción
DX(*) **DY(*)**	= coordenadas en el vídeo de la descripción
TF(*)	= matriz numérica con los códigos ASCII de las teclas de comando
D$(*)	= descripciones en la máscara vídeo
VO$(*)	= voces del menú principal
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de introducción alfanuméricos
NS$	= carácter "#"; identifica los campos de introducción numéricos
HO$	= borrado de la pantalla
NUC	= número de campos de introducción en la máscara vídeo activada
DS	= número de descripciones en la máscara vídeo
S$	= sigla del test considerado
H$(*)	= matriz de las cadenas para la ayuda
NL	= número de preguntas previstas
AA$	= cadena de I/O con el disco
MX	= último registro introducido
LR	= longitud registro

UP

MI(*)	= matriz de puntuaciones mínimas
MX(*)	= matriz de puntuaciones máximas (una gama de puntuación está constituida por un mínimo y un máximo)
B$(*)	= matriz de los diagnósticos finales
S$	= sigla del test considerado
DO	= indicador pregunta actual
R	= indicador respuesta dada
I	= índice bucle sobre las preguntas
MX	= último registro introducido
PP	= puntuación obtenida
PU(*)	= puntuaciones asociadas a las 10 preguntas
NF$	= fichero actual
A,B$,C,D	= gestión errores en disco

GA

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud del campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en el vídeo del campo de introducción
DX(*) **DY(*)**	= coordenadas en el vídeo de la descripción
TF(*)	= matriz numérica con los códigos ASCII de las teclas de comando
D$(*)	= descripciones en la máscara vídeo
VO$(*)	= voces del menú principal
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de introducción alfanuméricos
NS$	= carácter "#"; identifica los campos de introducción numéricos
HO$	= borrado de la pantalla
NUC	= número de campos de introducción en la máscara vídeo activada
DS	= número de descripciones en la máscara vídeo
H$(*)	= matriz de las cadenas para la ayuda
PA	= registro actual
S$	= sigla del test considerado
NF$	= nombre fichero datos actual
LR	= longitud registro
AA$	= cadena de I/O con el disco
MX	= último registro introducido

UA

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud del campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en el vídeo del campo de introducción
DX(*) **DY(*)**	= coordenadas en el vídeo de la descripción
TF(*)	= matriz numérica con los códigos ASCII de las teclas de comando
D$(*)	= descripciones en la máscara vídeo
VO$(*)	= voces del menú principal
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de introducción alfanuméricos
NS$	= carácter "#"; identifica los campos de introducción numéricos
HO$	= borrado de la pantalla
NUC	= número de campos de introducción en la máscara vídeo activada
DS	= número de descripciones en la máscara vídeo
NO$(*)	= matriz para los elementos
FO$(*)	= matriz de los punteros a atributos
DT$(*)	= matriz de atributos
S$	= sigla del test considerado
MX	= último registro
PA	= registro actual
ER	= indicador de error
R$	= respuesta

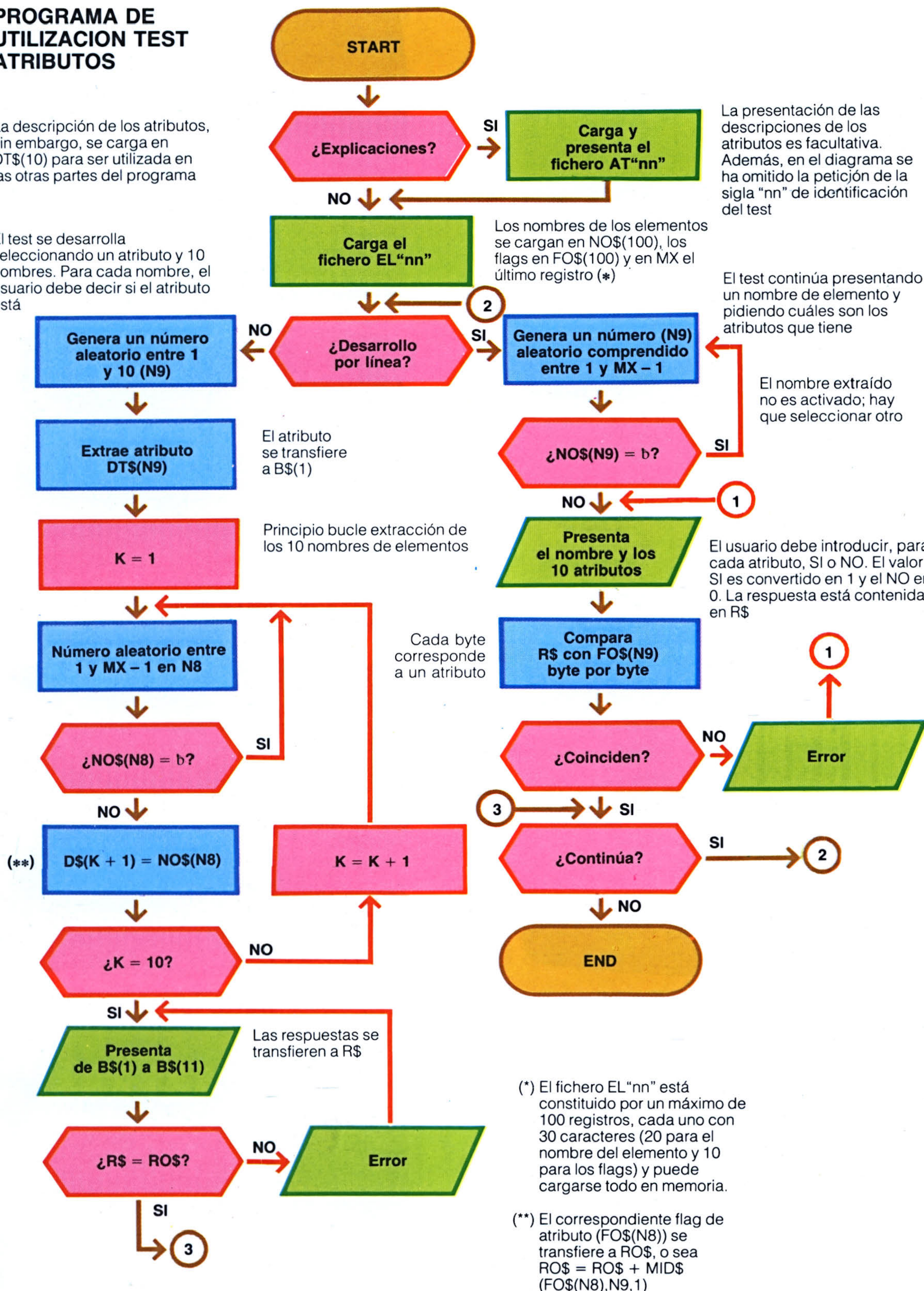
GS

SP$(*, *)	= matriz para los datos en forma alfanumérica de los 5 sprites
TA(*)	= teclas reconocidas
LZ(*)	= matriz de los bloques reservados a datos sprite
A,B$,C,D	= flags para errores en disco
PA	= registro actual
AA$	= cadena de I/O con el disco
NRM	= número máximo de introducciones
NS	= sprite actual
R	= respuesta prevista
MX	= última introducción
SP	= 1.er registro del VIC-II
P1	= posición de partida para los datos sprite
CA	= código de la tecla pulsada en creación sprite
FA	= índice en la matriz TA(*)

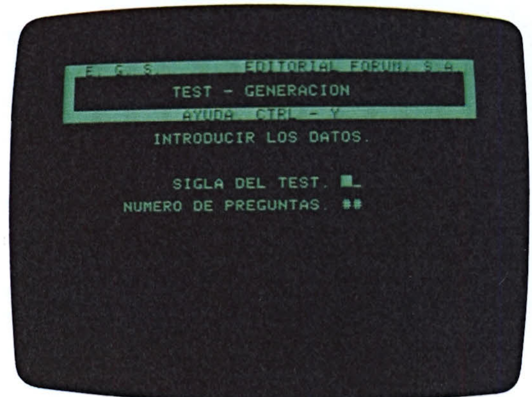
US

SP$(*, *)	= matriz para los datos de los 5 sprites
LZ(*)	= matriz de los bloques reservados a datos sprite
A,B$,C,D	= flags para errores en disco
MX	= último registro introducido
AA$	= cadena de I/O con el disco
RI	= respuesta introducida
R	= respuesta prevista
TE	= test actual
NF$	= fichero considerado
BL	= dirección de partida para los ficheros datos sprite
IN	= registro de partida de los datos test núm. TE

PROGRAMA DE UTILIZACION TEST ATRIBUTOS



Test con puntuación. La foto muestra la fase inicial de la generación de un test del tipo "con preguntas y puntuación" en la versión Apple. Obsérvese que el procedimiento HELP (ayuda) siempre está en línea o sea, es activo.



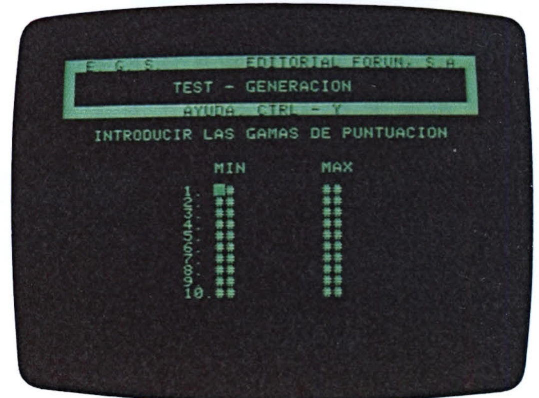
La siguiente máscara se utiliza para la introducción del texto que contiene las necesarias explicaciones.



Para cada pregunta (la foto corresponde a la n.° 2) deben indicarse los puntos asociados a todas las respuestas previstas. Estos puntos se irán sumando durante el desarrollo del test para dar el "juicio".



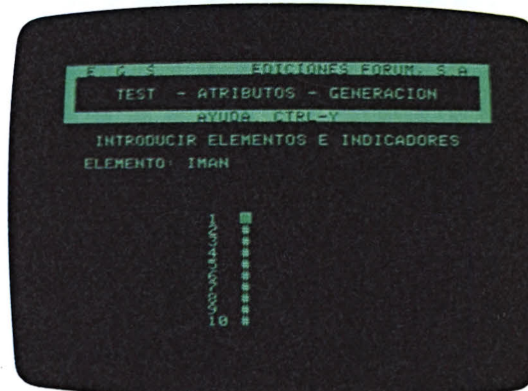
Como se ha indicado, la máquina emite el "juicio" en base a la puntuación alcanzada. La foto muestra la fase de definición de las "gamas" de puntuación.



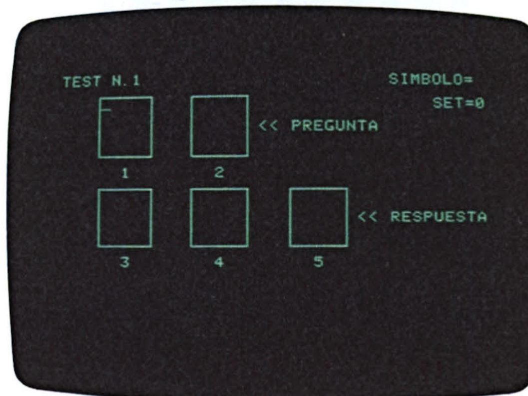
Test con atributos. En la foto se muestra la fase de generación que, como se ve, es completamente análoga a las de las otras versiones.



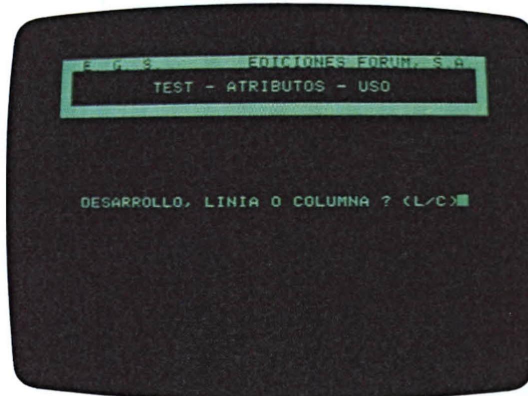
La máscara que sigue se utiliza para asociar a cada elemento (en la foto el imán) uno o más atributos definidos anteriormente. En esta versión, el símbolo "1" indicará que el atributo lo tiene el elemento. El símbolo "0", que el atributo le es ajeno.



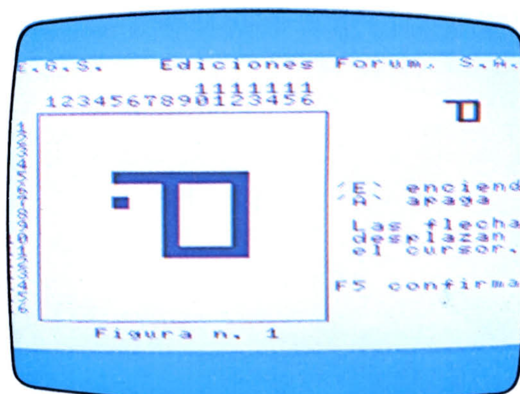
Test por símbolos. La lógica y la finalidad son idénticas a las de las otras versiones, mientras que el desarrollo del programa es notablemente distinto. De hecho, la ausencia de una gestión de los sprites ha obligado a desarrollar una data entry de alta resolución.



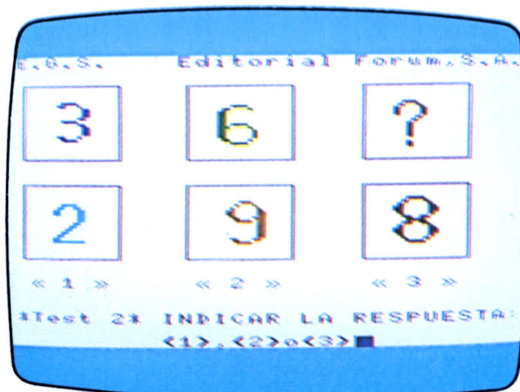
A pesar de la notable diversidad de los programas, el procedimiento conserva las mismas funcionalidades de los otros. La foto muestra la máscara vídeo utilizada para seleccionar el tipo de lógica que se desea seguir en el desarrollo de un test por atributos.



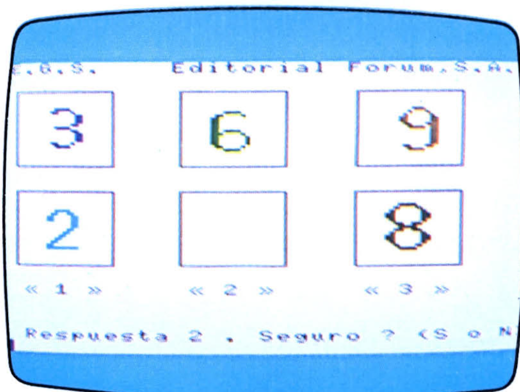
Test por símbolos. La foto muestra la máscara vídeo de la versión MSX utilizada en la fase de construcción de cada uno de los símbolos previstos en este test. En la misma máscara aparecen las breves notas explicativas necesarias para el empleo de esta función.



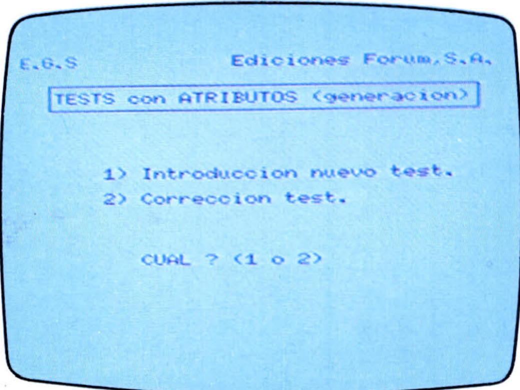
Terminada la fase de generación, se puede pasar al uso del test. En este caso, los símbolos utilizados son números y, por tanto, la finalidad del test es la de comprobar la capacidad de analizar algunas sencillas series numéricas.



La respuesta dada es correcta: efectivamente, el elemento necesario para completar la serie es 9.



Test con atributos. La foto muestra la máscara de selección de la función en la generación de un test con atributos.





Preguntas-Generación

```
10   REM  -------------------------
20   REM   TEST-GENERACION PREGUNTAS
30   REM          VERSION DOS
40   REM  -------------------------
50   REM   INICIALIZACIONES
60   DIM TF(15),TP(20),LC(20),ND(20),C
     X(20),CY(20),DX(20),DY(20),B$(20)
     ,D$(20)
70   BP$ =  CHR$ (7):
     NS$ = "#":
     AS$ =  CHR$ (95):
     PD$ = ".":
     BL$ = " ":
     D$ =  CHR$ (4):
     US$ =  CHR$ (13):
     ZE$ = "0":
     GIU$ = " ":
     SL$ = "/"
80   B$ = "                    ":
     BA$ = B$ + B$ + B$ + B$:
     B$ = ""
90   HE$ = "AYUDA. CTRL - Y"
100  HE$(1) = "CTRL Y. AYUDA ('ESPACIO
     ' PARA CONTINUAR)"
110  HE$(2) = "ESC. ANULA TODA LA INTR
     ODUCCION          "
120  HE$(3) = "RETURN. ALINEA Y CONVAL
     IDA INTRODUCCION "
130  HE$(4) = "CTRL I. PERMITE LA INSE
     RCION DE ESPACIOS"
140  HE$(5) = "CTRL D. BORRA UN CARACT
     ER                "
150  HE$(6) = "CTRL R. VUELVE AL MENU
     PRINCIPAL         "
160  HE$(7) = "CTRL W,A,S,Z.MUEVEN EN
     LAS 4 DIRECCIONES"
170  HE$(8) = "CTRL E. INTRODUCCION EN
      DISCO            "
172  TEXT :
     HOME :
     YT = 12:
     TI$ = "CUANTAS UNIDADES HAY ?(1/2
     )?":
     GOSUB 11200
174  GET A$:
     IF A$ < "1" OR A$ > "2" THEN 172
176  DR$ = A$:
     PRINT DR$
178  HOME :
     YT = 12:
     TI$ = "INTRODUCIR DISCO DATOS EN
     LA UNIDAD " + DR$:
     GOSUB 11200:
     GOSUB 11100
179  ONERR  GOTO 40000
180  MP$ = "TEST - GENERACION"
190  RESTORE :
     READ DA$,NV,A$,A$:
     FOR I = 1 TO 15:
       READ TF(I):
     NEXT I
195  DIM P$(20)
200  GOSUB 500
210  GOSUB 5000:
     IF FS = 7 THEN  GOSUB 6600:
     TEXT :
     HOME :
     END
215  GOSUB 8000:
     IF FS = 7 THEN  GOSUB 6600:
     GOTO 200
220  GOSUB 12000:
     IF FS = 7 THEN  GOSUB 6600:
     GOTO 215
230  GOSUB 14000:
     IF FS = 7 THEN  GOSUB 6600:
     GOTO 220
410  Y = 5:
     X = 1:
     GOSUB 11000:
     INVERSE :
     PRINT  LEFT$ (BA$,40);:
     NORMAL
500  REM  ----------------
510  REM  PRESENTA CABECERA
520  REM  ----------------
530  POKE 34,0:
     HOME :
     INVERSE :
     PRINT "  E. G. S. " +  LEFT$ (BA$
     ,7) + "EDITORIAL FORUN, S.A."
      + "  ";:
     NORMAL
540  YT = 3:
     TI$ = MP$:
     GOSUB 11200
550  Y = 2:
     X = 1:
     INVERSE :
     GOSUB 11000:
     PRINT " ": '
     X = 40:
     GOSUB 11000:
     PRINT " ";:
     Y = 3:
     X = 1:
     GOSUB 11000:
     PRINT " ":
     X = 40:
     GOSUB 11000:
     PRINT " ";:
     Y = 4:
     X = 1:
     GOSUB 11000:
     PRINT " ":
     X = 40:
     GOSUB 11000:
     PRINT " ";:
     Y = 5:
     X = 1:
     GOSUB 11000
560  PRINT  LEFT$ (BA$,40);:
     NORMAL :
     POKE 34,5
570  RETURN
2000 REM  ----------------
2010 REM  PRESENTA MASCARA
2020 REM  ----------------
2030 HOME :
     IF OPZ <  > 0 THEN Y = 7:
     X = 1:
     GOSUB 11000:
     PRINT " VOZ ";OPZ
2035 IF HE <  > 1 THEN YT = 5:
     TI$ = HE$:
     INVERSE :
     GOSUB 11200:
     NORMAL
2040 YT = 7:   ,
     TI$ = IN$:
     GOSUB 11200
2050 IF CO <  > 0 THEN Y = 7:
     X = XT +  LEN (IN$) + 1:
     GOSUB 11000:
     PRINT CO
2060 SP = 33 -  LEN (VO$(OPZ))
2070 SP$ = ""
2080 IF SP = 0 THEN 2100
2090 FOR J = 1 TO SP:
       SP$ = SP$ + BL$:
     NEXT
2100 LP =  LEN (PROG$)
2110 SI =  INT ((40 - LP) / 2):
     DE = 40 - LP - SI
2120 SI$ = "":
     DE$ = ""
2130 FOR J = 1 TO SI
2140   SI$ = SI$ + BL$
2150 NEXT
2160 FOR J = 1 TO DE
2170   DE$ = DE$ + BL$
2180 NEXT
2240 REM  MASCARA
2250 FOR J = 1 TO NUC
2260   A$ = ""
2270   X = DX(J):
       Y = DY(J):
       GOSUB 11000:
       :
       PRINT D$(J)
2280   NORMAL
2290   IF TP(J) <  > 5 AND TP(J)
        <  > 4 THEN 2330
2300   X = CX(J):
       Y = CY(J):
       GOSUB 11000
2310   PRINT B$(J)
2320   GOTO 2500
2330   N1 = LC(J):
       N2 = ND(J):
       IF TP(J) = 1 OR TP(J) = 2
        THEN C$ = NS$:
       GOTO 2350
2340   C$ = AS$
2345   IF N1 = 1 THEN A$ = C$:
       GOTO 2390
2350   FOR I = 1 TO N1 - N2 - 1:
         A$ = A$ + C$:
       NEXT I
2360   IF N2 = 0 THEN A$ = A$ + C$:
       GOTO 2390
2370   A$ = A$ + PD$
2380   FOR I = 1 TO N2:
         A$ = A$ + ZE$:
       NEXT I
2390   X = CX(J):
       Y = CY(J):
       GOSUB 11000
2400   PRINT A$
2410   B$(J) = ""
2420   FOR K = 1 TO N1 - N2 - 1
2430     B$(J) = B$(J) + BL$
2440   NEXT K
2450   IF N2 = 0 THEN B$(J) = B$(J)
        + BL$:
       GOTO 2500
2460   B$(J) = B$(J) + PD$
2470   FOR K = 1 TO N2
2480     B$(J) = B$(J) + BL$
2490   NEXT K
2500 NEXT J
2510 RETURN
2600 REM  ----------------
```

```
2610  REM  COMPLETA MASCARA
2620  REM  ----------------
2625  IF  LEFT$ (NF$,2) = "FT" THEN 265
      0
2630  TI$ = "INTRODUCIR LA PREGUNTA N."
       +  STR$ (JD):
      YT = 2:
      GOSUB 11200
2640  RETURN
2650  Y = 10:
      X = 16:
      GOSUB 11000:
      PRINT "MIN"
2660  X = 26:
      GOSUB 11000:
      PRINT "MAX"
2670  RETURN
3000  REM  ----------
3010  REM  DATA ENTRY
3020  REM  ----------
3030  J = 1
3040  NC = J:
      X = CX(NC):
      Y = CY(NC)
3050  CC = 1
3060  GOSUB 3240
3070  IF FS = 0 THEN 3100
3080  IF FS > 4 THEN 3210
3090  GOTO 3120
3100  J = J + 1:
      IF J > NUC THEN 3030
3110  GOTO 3040
3120  ON FS GOTO 3130,3150,3170,3190
3130  IF J > 1 THEN J = J - 1:
      GOTO 3040
3140  PRINT BP$;:
      FS = 0:
      J = NUC:
      GOTO 3040
3150  IF J < NUC THEN J = J + 1:
      GOTO 3040
3160  PRINT BP$;:
      FS = 0:
      GOTO 3030
3170  IF CC < LC(J) THEN S0 = 1:
      CC = CC + 1:
      X = X + 1:
      GOSUB 11000:
      GOTO 3060
3180  PRINT BP$:
      FS = 0:
      GOTO 3100
3190  IF CC > 1 THEN S0 =  - 1:
      CC = CC - 1:
      X = X - 1:
      GOSUB 11000:
      GOTO 3060
3200  PRINT BP$;:
      FS = 0:
      GOTO 3040
3210  IF FS > 8 THEN FS = 0:
      GOTO 3060
3220  IF FS = 8 THEN FS = 0:
      GOTO 3100
3230  RETURN
3240  REM  -------------------
3250  REM  LECTURA DE UN CAMPO
3260  REM  -------------------
3270  FS = 0
3280  IF TP(NC) = 4 THEN  RETURN
3290  IF TP(NC) = 2 THEN  GOSUB 9000:
      B$(NC) =  LEFT$ (B$(NC),LC(NC)):
      CC =  LEN (B$(NC)) + 1:
      GOSUB 4080:
      RETURN
3300  IF  MID$ (B$(NC),CC,1) = PD$
       THEN X = X + S0:
      CC = CC + S0
3310  GOSUB 11000
3320  GET A$:
      IF A$ = "" THEN 3320
3330  CA =  ASC (A$):
      IF CA > 31 THEN 3380
3340  FOR I = 1 TO 15
3350     IF TF(I) = (CA) THEN FS
          = I:
         GOSUB 3630:
         RETURN
3360  NEXT I
3370  GOTO 3320
3380  IF TP(NC) <  > 1 THEN 3440
3390  IF CA = 45 OR CA = 46 OR (CA
       >  = 48 AND CA <  = 57) THEN 341
      0
3400  PRINT BP$;:
      GOTO 3320
3410  IF CA <  > 46 THEN 3440
3420  GOSUB 4270
3430  IF SWP = 1 THEN 3400
3440  S0 = 1:
      GOSUB 3520
3450  IF CC = LC(NC) THEN  RETURN
3460  CC = CC + 1:
      X = X + 1:
      GOTO 3300
3520  REM  -----------------------
3530  REM  TRANSFERENCIA AL BUFFER
3540  REM  -----------------------
3550  IF CC = 1 THEN SN$ = "":
      GOTO 3570
3560  SN$ =  LEFT$ (B$(NC),CC - 1)
3570  IF CC = LC(NC) THEN DS$ = "":
      GOTO 3590
3580  N = LC(NC) - CC:
      DS$ =  RIGHT$ (B$(NC),N)
3590  B$(NC) = SN$ + A$ + DS$
3620  PRINT A$:
      RETURN
3630  REM  --------
3640  REM, COMANDOS
3650  REM  --------
3660  IF FS >  = 5 THEN 3690
3670  IF FS = 5 AND TP(NC) = 1 THEN
       GOSUB 4080
3680  RETURN
3690  IF FS = 9 THEN  GOSUB 13000:
      RETURN
3700  IF FS = 8 THEN  GOSUB 4080:
      RETURN
3710  IF FS < 8 THEN  RETURN
3720  ON FS - 9 GOSUB 3740,3910
3730  RETURN
3740  REM  ---------
3750  REM  INSERCION
3760  REM  ---------
3770  IF CC = LC(NC) THEN  PRINT BP$:
      RETURN
3780  IF CC = 1 THEN SN$ = "":
      GOTO 3800
3790  SN$ =  LEFT$ (B$(NC),CC - 1)
3800  N = LC(NC) - CC + 1:
      DS$ =  RIGHT$ (B$(NC),N)
3820  C$ = BL$
3830  A$ = SN$ + C$ + DS$
3840  B$(NC) =  LEFT$ (A$,LC(NC))
3850  XX = X:
      YY = Y
3860  X = CX(NC):
      Y = CY(NC):
      GOSUB 11000
3870  X = XX:
      Y = YY
3880  PRINT B$(NC):
      S0 =  - 1
3890  IF  RIGHT$ (B$(NC),1) = PD$
       THEN 3780
3900  RETURN
3910  REM  -------
3920  REM  BORRADO
3930  REM  -------
3940  IF TP(NC) <  > 1 THEN 3960
3950  IF  MID$ (B$(NC),2,1) = PD$
       AND CC = 1 THEN  RETURN
3960  IF CC < 1 THEN  PRINT BP$:
      RETURN
3970  IF CC = LC(NC) THEN DS$ = "":
      GOTO 4000
3980  N = LC(NC) - CC:
      DS$ =  RIGHT$ (B$(NC),N)
3990  IF CC = 1 THEN SN$ = "":
      GOTO 4010
4000  SN$ =  LEFT$ (B$(NC),CC - 1)
4010  IF TP(NC) = 1 OR TP(NC) = 2
       THEN C$ = NS$:
      GOTO 4030
4020  C$ = AS$
4030  B$(NC) = SN$ + DS$ + C$
4040  XX = X:
      YY = Y
4050  X = CX(NC):
      Y = CY(NC):
      GOSUB 11000
4060  X = XX:
      Y = YY
4070  PRINT B$(NC):
      S0 = 1:
      RETURN
4080  REM  ---------
4090  REM  ALINEADOS
4100  REM  ---------
4110  IF LC(NC) = 1 THEN  RETURN
4120  N1 = LC(NC) - CC + 1
4130  IF CC = 1 THEN B$(NC) = "":
      GOTO 4150
4140  B$(NC) =  LEFT$ (B$(NC),CC
       - 1)
4150  IF TP(NC) = 1 OR TP(NC) = 2
       THEN 4200
4160  FOR I = 1 TO N1
4170     B$(NC) = B$(NC) + BL$
4180  NEXT I
4190  GOTO 4240
4200  FOR I = 1 TO N1
4210     B$(NC) = BL$ + B$(NC)
4220  NEXT I
4230  IF  RIGHT$ (B$(NC),1) = PD$
       THEN B$(NC) = BL$ +  LEFT$ (B$(N
      C),LC(NC) - 1)
4240  X = CX(NC):
      Y = CY(NC)
4250  GOSUB 11000
4260  PRINT B$(NC):
      RETURN
4270  REM  ----------
4280  REM  CTRL.PUNTO
4290  REM  ----------
4300  SWP = 0
4310  IF CC = 1 OR CC = LC(NC) THEN SWP
       = 1:
      RETURN
4320  FOR I = 1 TO LC(NC)
4330     IF  MID$ (B$(NC),I,1) = PD$
          THEN I = LC(NC):
         SWP = 1:
         RETURN
4340  NEXT I:
      RETURN
5000  REM  -----------------------
5010  REM  INT.SIGLA Y N.PREGUNTAS
5020  REM  -----------------------
5030  LM = 0:
      GOSUB 7000
5040  IN$ = "INTRODUCIR LOS DATOS.":
      C0 = 0:
      GOSUB 2000
5050  GOSUB 3000:
      IF FS = 6 THEN 5040
```

```
5060  IF FS = 7 THEN  RETURN
5070  NO =  VAL (B$(2)):
      IF NO <  = 0 OR NO > 20 THEN
       GOSUB 1900:
      TP(1) = 4:
      TP(2) = 4:
      GOSUB 2000:
      GOSUB 7000:
      GOTO 5050
5075  N$ = B$(1)
5080  NF$ = "FP" + N$:
      LR = 35:
      GOSUB 6000
5085  NR = 1:
      AA$ = B$(2):
      GOSUB 6400
5090  RETURN
6000  REM  ----------------
6010  REM  APERTURA FICHERO
6020  REM  ----------------
6030  PRINT D$
6040  PRINT D$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D$
6060  RETURN
6200  REM  ----------------
6210  REM  LECTURA REGISTRO
6220  REM  ----------------
6230  PRINT D$
6240  PRINT D$;"READ";NF$;",R";NR
6250  INPUT AA$
6260  PRINT D$
6270  RETURN
6400  REM  ------------------
6410  REM  ESCRITURA REGISTRO
6420  REM  ------------------
6430  PRINT D$
6440  PRINT D$;"WRITE";NF$;",R";NR
6450  PRINT  CHR$ (34) + AA$
6460  PRINT D$
6470  RETURN
6600  REM  --------------
6610  REM  CIERRA FICHERO
6620  REM  --------------
6630  PRINT D$
6640  PRINT D$;"CLOSE"
6650  PRINT D$
6660  RETURN
7000  REM  --------------------
7010  REM  LECTURA DE LOS DATOS
7020  REM  --------------------
7030  RESTORE
7040  READ DA$:
      READ NV
7050  FOR I = 1 TO NV + 15:
         READ DA$:
      NEXT I
7060  IF LM = 0 THEN 7075
7070  ND = 0:
      FOR I = 0 TO LM - 1:
         READ DA:
         ND = ND + DA + (DA * 7):
      NEXT I
7075  READ NUC:
      IF NUC = 0 THEN  RETURN
7080  IF NV - LM - 1 = 0 THEN 7090
7085  FOR I = 1 TO NV - LM - 1:
         READ DA$:
      NEXT I
7090  IF LM = 0 THEN 7100
7092  IF ND = 0 THEN 7100
7095  FOR I = 1 TO ND:
         READ DA$:
      NEXT I
7100  FOR I = 1 TO NUC:
         READ D$(I):
      NEXT I
7110  FOR I = 1 TO NUC:
         READ TP(I),LC(I),ND(I),DX(I),D
         Y(I),CX(I),CY(I):
      NEXT I
7120  RETURN
8000  REM  ----------------------
8010  REM  INTRODUCCION PREGUNTAS
8020  REM  ----------------------
8030  ID = 2:
      JD = 1:
      NUC = 20
8040  FOR I = 1 TO 20:
         TP(I) = 3:
         ND(I) = 0:
         LC(I) = 35:
         D$(I) = "":
         DX(I) = 1:
         DY(I) = 1:
         CX(I) = 4:
         CY(I) = I + 3:
      NEXT I
8050  IN$ = "":
      HE = 1
8060  POKE 34,0:
      GOSUB 2000:
      GOSUB 2600
8070  GOSUB 3000:
      IF FS = 6 THEN 8060
8080  IF FS = 7 THEN  RETURN
8090  FOR I = ID TO ID + 19:
         NR = I
8100     AA$ = B$(I - ID + 1):
         GOSUB 6400
8110  NEXT I
8120  IF JD < NO THEN JD = JD + 1:
      ID = ID + 20:
      GOTO 8050
8130  GOSUB 6600:
      GOSUB 500:
      HE = 0:
      RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA "TESTS - GEN."
10040 DATA 2,"",""
10050 DATA 23,26,19,1,5,27,18,13,25,9,4
      ,0,0,0,0
10060 DATA 2,2
10070 DATA "SIGLA DEL TEST.","NUMERO DE
       PREGUNTAS."
10080 DATA 3,2,0,12,11,28,11
10090 DATA 1,2,0,7,13,28,13
10100 DATA "RESULTADOS.",""
10110 DATA 3,40,0,16,11,1,14
10120 DATA 3,40,0,1,1,1,16
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 HTAB X:
      VTAB Y
11040 RETURN
11100 REM ---------------
11110 REM ESPERA CARACTER
11120 REM ---------------
11130 GOSUB 11600:
      YT = 23:
      TI$ = "PULSAR UNA TECLA":
      FLASH :
      GOSUB 11200
11140 PRINT BP$;:
      X = 20:
      Y = 23:
      GOSUB 11000:
      GET A$:
      GOSUB 11600
11150 RETURN
11200 REM ---------------
11210 REM CENTRA EN VIDEO
11220 REM ---------------
11230 IF TI$ = "" THEN  RETURN
11235 TI$ =  LEFT$ (TI$,40)
11240 XT = (42 -  LEN (TI$)) / 2
11250 HTAB XT:
      VTAB YT
11260 PRINT TI$;:
      RETURN
11600 REM ------------------
11610 REM BORRA ULTIMA LINEA
11620 REM ------------------
11630 Y = 23:
      X = 1:
      GOSUB 11000
11640 NORMAL :
      PRINT  LEFT$ (BA$,40);:
      Y = 10:
      GOSUB 11000
11650 RETURN
12000 REM ----------------------
12010 REM INTRODUCCION RESPUESTA
12020 REM ----------------------
12030 NF$ = "FR" + N$:
      LR = 35:
      GOSUB 6000
12040 NUC = 20:
      J = 11:
      FOR I = 1 TO 20 STEP 2
12050    TP(I) = 3:
         TP(I + 1) = 1:
         LC(I) = 33
12060    LC(I + 1) = 2:
         D$(I) = "":
         D$(I + 1) = ""
12070    DX(I) = 1:
         DX(I + 1) = 1:
         CX(I) = 2
12080    CX(I + 1) = 37:
         CY(I) = J:
         CY(I + 1) = J
12090    J = J + 1:
      NEXT I:
      D$(1) = "RESPUESTA":
      D$(2) = "PUNTOS":
      DX(1) = 2:
      DY(1) = 9:
      DX(2) = 35:
      DY(2) = 9
12100 IR = 1:
      FOR JR = 1 TO NO
12110    IN$ = "RESPUESTA A LA PREGUNTA
          N." +  STR$ (JR):
         CO = 0
12120    GOSUB 2000:
         GOSUB 3000
12130    IF FS = 6 THEN 12120
12140    IF FS = 7 THEN  RETURN
12150    J = 1:
         FOR I = IR TO IR + 9
12160       NR = I:
            AA$ = B$(J) + B$(J + 1):
            GOSUB 6400
12170       J = J + 2:
         NEXT I:
         IR = IR + 10
12180 NEXT JR
12190 GOSUB 6600:
      RETURN
13000 REM -----
13010 REM AYUDA
13020 REM -----
13030 XX = X:
      YY = Y
13040 INVERSE
13050 K = 1
13060 X = 1:
      Y = 5:
      GOSUB 11000
13070 PRINT HE$(K);
13080 B =  PEEK ( - 16384)
13090 POKE  - 16368,0
```

```
13100 IF B < 128 THEN 13080
13110 IF B <  > 160 THEN 13080
13120 K = K + 1:
      IF K > 8 THEN 13140
13130 GOTO 13060
13140 Y = 5:
      X = 1:
      GOSUB 11000:
      PRINT  LEFT$ (BA$,40);
13150 YT = 5:
      TI$ = HE$:
      GOSUB 11200
13160 NORMAL
13170 X = XX:
      Y = YY
13180 RETURN
14000 REM -----------------------
14010 REM INT.GAMAS DE PUNTUACION
14020 REM -----------------------
14030 NF$ = "FT" + N$:
      LR = 80:
      GOSUB 6000
14040 NUC = 20:
      J = 12:
      FOR I = 1 TO 20 STEP 2
14050    LC(I) = 2:
         LC(I + 1) = 2:
         TP(I) = 1:
         TP(I + 1) = 1
14060    CX(I) = 16:
         CX(I + 1) = 26:
         CY(I) = J:
         CY(I + 1) = J
14065    D$(I) =  STR$ (J - 11) + ".":
         D$(I + 1) = "":
         DX(I) = 13:
         DX(I + 1) = 1:
         DY(I) = J:
         DY(I + 1) = 1
14070    J = J + 1:
      NEXT I
14075 IN$ = "INTRODUCIR LAS GAMAS DE PU
      NTUACION"
14080 GOSUB 2000:
      GOSUB 2600
14090 GOSUB 3000:
      IF FS = 6 THEN 14080
14100 IF FS = 7 THEN  RETURN
14110 NR = 1:
      AA$ = ""
14120 FOR I = 1 TO 20:
         AA$ = AA$ + B$(I):
         P$(I) = B$(I):
      NEXT I
14130 GOSUB 6400
14140 LM = 1:
      GOSUB 7000
14150 JD = 1:
      FOR ID = 1 TO 10
14160    IN$ = "PUNTUACION " + P$(JD)
          + " - " + P$(JD + 1)
14170    GOSUB 2000:
         GOSUB 3000
14180    IF FS = 6 THEN 14170
14190    IF FS = 7 THEN  RETURN
14200    NR = ID + 1:
         AA$ = B$(1) + B$(2):
         GOSUB 6400
14210    JD = JD + 2:
      NEXT ID
14220 GOSUB 6600:
      HOME :
      YT = 13
14230 TI$ = "GENERACION EFECTUADA":
      INVERSE
14240 GOSUB 11200:
      NORMAL :
      PRINT :
      PRINT :
      END
40000 REM -------------
40010 REM GESTION ERROR
40020 REM -------------
40025 TEXT :
      HOME
40030 IF  PEEK (222) = 5 THEN 40130
40040 IF  PEEK (222) <  > 8 THEN 40060
40050 YT = 13:
      TI$ = "COMPROBAR EL DISCO DATOS":
      INVERSE :
      GOSUB 11200:
      PRINT :
      END
40060 IF  PEEK (222) <  > 13 AND
       PEEK (222) <  > 11 THEN 40080
40070 YT = 13:
      TI$ = "NOMBRE FICHERO NO VALIDO":
      INVERSE :
      GOSUB 11200:
      PRINT :
      END
40080 IF  PEEK (222) <  > 9 THEN 40100
40090 YT = 13:
      TI$ = "DISCO LLENO":
      INVERSE :
      GOSUB 11200:
      PRINT :
      END
40100 IF  PEEK (222) <  > 4 THEN 40115
40110 YT = 13:
      TI$ = "DISCO PROTEGIDO":
      INVERSE :
      GOSUB 11200:
      PRINT :
      END
40115 L =  PEEK (218) + 256 *  PEEK (21
      9)
40120 HOME :
      YT = 13:
      TI$ = "ERROR " +  STR$ ( PEEK (22
      2)) + " EN " +  STR$ (L):
      GOSUB 11200:
      END
40130 TI$ = "EL NOMBRE DEL FICHERO NO E
      XISTE":
      YT = 13:
      INVERSE :
      GOSUB 11200:
      PRINT :
      END
```

Preguntas-Uso

```
10    REM   ------------------
20    REM   TEST-PREGUNTAS-USO
30    REM      VERSION  DOS
40    REM   ------------------
50    REM   INICIALIZACIONES
60    DIM B$(20):
      BP$ =  CHR$ (7):
      D$ =  CHR$ (4)
70    B$ = "                    "
80    BA$ = B$ + B$ + B$ + B$:
      B$ = ""
172   TEXT :
      HOME :
      YT = 12:
      TI$ = "CUANTAS UNIDADES HAY ?(1/2
      )":
      GOSUB 11200
174   GET A$:
      IF A$ < "1" OR A$ > "2" THEN 172
176   DR$ = A$:
      PRINT DR$
178   HOME :
      YT = 12:
      TI$ = "INTRODUCIR DISCO DATOS EN
      LA UNIDAD " + DR$:
      GOSUB 11200:
      GOSUB 11100
179   ONERR  GOTO 40000
180   MP$ = "TEST  - USO"
200   GOSUB 500
210   GOSUB 5000
220   HOME :
      GOTO 210
500   REM   -----------------
510   REM   PRESENTA CABECERA
520   REM   -----------------
530   POKE 34,0:
      HOME :
      INVERSE :
      PRINT "  E. G. S. " +  LEFT$ (BA$
      ,7) + "EDITORIAL FORUM, S.A."
       + "  ";:
      NORMAL
540   YT = 3:
      TI$ = MP$:
      GOSUB 11200
550   Y = 2:
      X = 1:
      INVERSE :
      GOSUB 11000:
      PRINT " ":
      X = 40:
      GOSUB 11000:
      PRINT " ";:
      Y = 3:
      X = 1:
      GOSUB 11000:
      PRINT " ":
      X = 40:
      GOSUB 11000:
      PRINT " ";:
      Y = 4:
      X = 1:
      GOSUB 11000:
      PRINT " ":
      X = 40:
      GOSUB 11000:
      PRINT " ";:
      Y = 5:
      X = 1:
      GOSUB 11000
560   PRINT  LEFT$ (BA$,40);:
      NORMAL :
      POKE 34,5
570   RETURN
5000  REM  -----
5010  REM  START
5020  REM  -----
5030  YT = 7:
      TI$ = "INTRODUCIR LA SIGLA DEL TE
      ST":
      GOSUB 11200
5040  X = 16:
      Y = 13:
      GOSUB 11000:
      PRINT "SIGLA "; CHR$ (95);
       CHR$ (95);:
      X = 21:
      GOSUB 11000
5050  INPUT S$:
      IF S$ = "" THEN 5040
5060  S$ =  LEFT$ (S$,2):
      IF  LEN (S$) < 2 THEN S$ = S$
       + " "
5070  NF$ = "FP" + S$:
      LR = 35:
      GOSUB 6000
5080  E = 1:
      NR = 1:
      GOSUB 6200:
      NO =  VAL (AA$):
      E = 0
5090  ID = 2:
      IR = 1:
```

```
     PP = 0
5100 JD = 1
5110 FOR I = ID TO ID + 19:
        NR = I
5120    GOSUB 6200:
        B$(I - ID + 1) = AA$
5130 NEXT I:
     POKE 34,0
5140 HOME :
     YT = 1:
     TI$ = "PREGUNTA N." +  STR$ (JD):
     INVERSE :
     GOSUB 11200
5145 NORMAL
5150 X = 4:
     FOR Y = 2 TO 21:
        GOSUB 11000
5160    PRINT B$(Y - 1):
     NEXT Y
5170 GOSUB 11100:
     GOSUB 500
5180 GOSUB 6600:
     NF$ = "FR" + S$:
     LR = 35:
     GOSUB 6000
5190 FOR I = IR TO IR + 9:
        NR = I:
        GOSUB 6200
5200    B$(I - IR + 1) =  LEFT$ (AA$,3
        3)
5210    P(I - IR + 1) =  VAL ( MID$ (A
        A$,34,2))
5220 NEXT I
5230 YT = 7:
     TI$ = "ELEGIR UNA DE LAS SIGUIENT
     ES SOLUCIONES":
     GOSUB 11200
5240 X = 3:
     FOR Y = 10 TO 19:
        GOSUB 11000
5250    PRINT Y - 10;". ";B$(Y - 9)
5260 NEXT Y
5270 Y = 22:
     GOSUB 11000:
     PRINT "CUAL ?";
5280 GET A$:
     IF A$ < "0" OR A$ > "9" THEN 5270

5290 I =  VAL (A$):
     PP = PP + P(I + 1)
5300 IF JD < NO THEN  GOSUB 6600:
     NF$ = "FP" + S$:
     LR = 35:
     GOSUB 6000:
     JD = JD + 1:
     IR = IR + 10:
     ID = ID + 20:
     GOTO 5110
5310 GOSUB 6600
5320 NF$ = "FT" + S$:
     LR = 80:
     GOSUB 6000
5330 NR = 1:
     GOSUB 6200
5340 J = 1:
     FOR I = 1 TO 40 STEP 4
5350    MI(J) =  VAL ( MID$ (AA$,I,2))
5360    MX(J) =  VAL ( MID$ (AA$,I
        + 2,2))
5370    J = J + 1:
     NEXT I
5380 FOR I = 1 TO 10
5390    IF PP > = MI(I) AND PP
        < = MX(I) THEN 5420
5400 NEXT I
5410 HOME :
     YT = 13:
     TI$ = "ERROR DE GENERACION TEST":
     INVERSE :
     GOSUB 11200:
     END
5420 NR = I + 1:
     GOSUB 6200:
     B$(1) =  LEFT$ (AA$,40)
5430 B$(2) =  MID$ (AA$,41,40)
5440 HOME :
     YT = 9:
     TI$ = "RESULTADOS.":
     INVERSE :
     GOSUB 11200:
     NORMAL
5450 PRINT :
     PRINT :
     PRINT :
     PRINT B$(1):
     PRINT B$(2)
5460 X = 12:
     Y = 21:
     GOSUB 11000:
     PRINT "OTRO TEST ? (S/N)";
5470 GET A$:
     IF A$ < > "S" AND A$ < > "N"
      THEN 5460
5480 IF A$ = "S" THEN  RETURN
5490 TEXT :
     HOME :
     END
6000 REM  ----------------
6010 REM  APERTURA FICHERO
6020 REM  ----------------
6030 PRINT D$
6040 PRINT D$;"OPEN";NF$ + ",D"
      + DR$;",L";LR + 2
6050 PRINT D$
6060 RETURN
6200 REM  ----------------
6210 REM  LECTURA REGISTRO
6220 REM  ----------------
6230 PRINT D$
6240 PRINT D$;"READ";NF$;",R";NR
6250 INPUT AA$
6260 PRINT D$
6270 RETURN
6600 REM  --------------
6610 REM  CIERRA FICHERO
6620 REM  --------------
6630 PRINT D$
6640 PRINT D$;"CLOSE";NF$
6650 PRINT D$
6660 RETURN
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 HTAB X:
      VTAB Y
11040 RETURN
11100 REM ---------------
11110 REM ESPERA CARACTER
11120 REM ---------------
11130 GOSUB 11600:
      YT = 23:
      TI$ = "PULSAR UNA TECLA":
      FLASH :
      GOSUB 11200
11140 PRINT BP$;:
      X = 20:
      Y = 23:
      GOSUB 11000:
      GET A$:
      GOSUB 11600
11150 RETURN
11200 REM ---------------
11210 REM CENTRA EN VIDEO
11220 REM ---------------
11230 IF TI$ = "" THEN  RETURN
11235 TI$ =  LEFT$ (TI$,40)
11240 XT = (42 -  LEN (TI$)) / 2
11250 HTAB XT:
      VTAB YT
11260 PRINT TI$;:
      RETURN
11600 REM ------------------
11610 REM BORRA ULTIMA LINEA
11620 REM ------------------
11630 Y = 23:
      X = 1:
      GOSUB 11000
11640 NORMAL :
      PRINT  LEFT$ (BA$,40);:
      Y = 10:
      GOSUB 11000
11650 RETURN
40000 REM -------------
40010 REM GESTION ERROR
40020 REM -------------
40025 TEXT :
      HOME
40030 IF  PEEK (222) = 5 THEN 40130
40040 IF  PEEK (222) < > 8 THEN 40115
40050 YT = 13:
      TI$ = "COMPROBAR EL DISCO DATOS":
      INVERSE :
      GOSUB 11200:
      PRINT :
      END
40115 L =  PEEK (218) + 256 *  PEEK (21
      9)
40120 HOME :
      YT = 13:
      TI$ = "ERROR " +  STR$ ( PEEK (22
      2)) + " EN " +  STR$ (L):
      GOSUB 11200:
      END
40130 IF E = 1 THEN YT = 13:
      TI$ = "EL TEST NO EXISTE":
      INVERSE :
      GOSUB 11200:
      GOSUB 11100:
      GOTO 200
40140 YT = 13:
      TI$ = "LOS DATOS NO EXISTEN":
      INVERSE :
      GOSUB 11200:
      NORMAL :
      END
```

Atributos-Generación

```
10    REM    -------------------------
20    REM    TEST-GENERACION ATRIBUTOS
30    REM             VERSION DOS
40    REM    -------------------------
50    REM    INICIALIZACIONES
60    DIM TF(15),TP(20),LC(20),ND(20),C
      X(20),CY(20),DX(20),DY(20),B$(20)
      ,D$(20)
70    BP$ =  CHR$ (7):
      NS$ = "#":
      AS$ =  CHR$ (95):
      PD$ = ".":
      BL$ = " ":
      D$ =  CHR$ (4):
      US$ =  CHR$ (13):
      ZE$ = "0":
      GIU$ = " ":
      SL$ = "/"
80    B$ = "                            ":
      BA$ = B$ + B$ + B$ + B$:
      B$ = ""
90    HE$ = "AYUDA. CTRL-Y "
100   HE$(1) = "CTRL Y.AYUDA ('ESPACIO'
       PARA CONTINUAR) "
110   HE$(2) = "ESC. ANULA TODA LA INTR
      ODUCCION           "
120   HE$(3) = "RETURN. ALINEA Y CONVAL
      IDA INTRODUCCION "
130   HE$(4) = "CTRL I. PERMITE LA INSE
```

```
       RCION DE ESPACIOS"
140    HE$(5) = "CTRL D. BORRA UN CARACT
       ER         "
150    HE$(6) = "CTRL R. VUELVE AL MENU
       PRINCIPAL       "
160    HE$(7) = "CTRL W,A,S,Z.MUEVEN EN
       LAS 4 DIRECCIONES"
170    HE$(8) = "CTRL E. INTRODUCCION EN
        DISCO          "
172    TEXT :
       HOME :
       YT = 12:
       TI$ = "CUANTAS UNIDADES HAY ?( (1
       /2)":
       GOSUB 11200
174    GET A$:
       IF A$ < "1" OR A$ > "2" THEN 172
176    DR$ = A$:
       PRINT DR$
178    HOME :
       YT = 12:
       TI$ = "INTRODUCIR DISCO DATOS EN
       LA UNIDAD " + DR$:
       GOSUB 11200:
       GOSUB 11100
179    ONERR  GOTO 40000
180    MP$ = "TEST  - ATRIBUTOS - GENERA
       CION"
190    RESTORE :
       FOR I = 1 TO 15:
          READ TF(I):
       NEXT I
200    GOSUB 500
210    GOSUB 5000
215    IF FS = 7 THEN  GOSUB 6600:
       GOTO 250
220    GOSUB 8000
225    IF FS = 7 THEN  GOSUB 6600:
       GOTO 210
230    GOSUB 12000
235    IF FS = 7 THEN  GOSUB 6600:
       GOTO 220
250    TEXT :
       HOME :
       END
410    Y = 5:
       X = 1:
       GOSUB 11000:
       INVERSE :
       PRINT  LEFT$ (BA$,40);:
       NORMAL
500    REM  -----------------
510    REM  PRESENTA CABECERA
520    REM  -----------------
530    POKE 34,0:
       HOME :
       INVERSE :
       PRINT "  E. G. S. " +  LEFT$ (BA$
       ,7) + "EDICIONES FORUM, S.A."
        + "  ";:
       NORMAL
540    YT = 3:
       TI$ = MP$:
       GOSUB 11200
550    Y = 2:
       X = 1:
       INVERSE :
       GOSUB 11000:
       PRINT " ":
       X = 40:
       GOSUB 11000:
       PRINT " ";:
       Y = 3:
       X = 1:
       GOSUB 11000:
       PRINT " ":
       X = 40:
       GOSUB 11000:
       PRINT " ";:
       Y = 4:
       X = 1:
       GOSUB 11000:
       PRINT " ":
       X = 40:
       GOSUB 11000:
          PRINT " ";:
          Y = 5:
          X = 1:
          GOSUB 11000
560       PRINT  LEFT$ (BA$,40);:
          NORMAL :
          POKE 34,5
570       RETURN
2000   REM  ----------------
2010   REM  PRESENTA MASCARA
2020   REM  ----------------
2030   HOME :
          IF OPZ <  > 0 THEN Y = 7:
          X = 1:
          GOSUB 11000:
          PRINT " VOZ ";OPZ
2035   IF HE <  > 1 THEN YT = 5:
          TI$ = HE$:
          INVERSE :
          GOSUB 11200:
          NORMAL
2040   YT = 7:
          TI$ = IN$:
          GOSUB 11200
2050   IF CO <  > 0 THEN Y = 7:
          X = XT +  LEN (IN$) + 1:
          GOSUB 11000:
          PRINT CO
2060   SP = 33 -  LEN (VO$(OPZ))
2070   SP$ = ""
2080   IF SP = 0 THEN 2100
2090   FOR J = 1 TO SP:
             SP$ = SP$ + BL$:
          NEXT
2100   LP =  LEN (PROG$)
2110   SI =  INT ((40 - LP) / 2):
          DE = 40 - LP - SI
2120   SI$ = "":
          DE$ = ""
2130   FOR J = 1 TO SI
2140      SI$ = SI$ + BL$
2150   NEXT
2160   FOR J = 1 TO DE
2170      DE$ = DE$ + BL$
2180   NEXT
2240   REM  MASCARA
2250   FOR J = 1 TO NUC
2260      A$ = ""
2270      X = DX(J):
          Y = DY(J):
          GOSUB 11000:
          :
          PRINT D$(J)
2280      NORMAL
2290      IF TP(J) <  > 5 AND TP(J)
           <  > 4 THEN 2330
2300      X = CX(J):
          Y = CY(J):
          GOSUB 11000
2310      PRINT B$(J)
2320      GOTO 2500
2330      N1 = LC(J):
          N2 = ND(J):
          IF TP(J) = 1 OR TP(J) = 2
           THEN C$ = NS$:
          GOTO 2345
2340      C$ = AS$
2345      IF N1 = 1 THEN A$ = C$:
          GOTO 2390
2350      FOR I = 1 TO N1 - N2 - 1:
             A$ = A$ + C$:
          NEXT I
2360      IF N2 = 0 THEN A$ = A$ + C$:
             GOTO 2390
2370         A$ = A$ + PD$
2380         FOR I = 1 TO N2:
                A$ = A$ + ZE$:
             NEXT I
2390         X = CX(J):
             Y = CY(J):
             GOSUB 11000
2400         PRINT A$
2410         B$(J) = ""
2420         FOR K = 1 TO N1 - N2 - 1
2430            B$(J) = B$(J) + BL$
2440         NEXT K
2450         IF N2 = 0 THEN B$(J) = B$(J)
              + BL$:
             GOTO 2500
2460         B$(J) = B$(J) + PD$
2470         FOR K = 1 TO N2
2480            B$(J) = B$(J) + BL$
2490         NEXT K
2500   NEXT J
2510   RETURN
2600   REM  -----------------
2610   REM  COMPLETA MASCHERA
2620   REM  -----------------
2630   RETURN
3000   REM  ----------
3010   REM  DATA ENTRY
3020   REM  ----------
3030   J = 1
3040   NC = J:
       X = CX(NC):
       Y = CY(NC)
3050   CC = 1
3060   GOSUB 3240
3070   IF FS = 0 THEN 3100
3080   IF FS > 4 THEN 3210
3090   GOTO 3120
3100   J = J + 1:
       IF J > NUC THEN 3030
3110   GOTO 3040
3120   ON FS GOTO 3130,3150,3170,3190
3130   IF J > 1 THEN J = J - 1:
       GOTO 3040
3140   PRINT BP$;:
       FS = 0:
       J = NUC:
       GOTO 3040
3150   IF J < NUC THEN J = J + 1:
       GOTO 3040
3160   PRINT BP$;:
       FS = 0:
       GOTO 3030
3170   IF CC < LC(J) THEN SO = 1:
       CC = CC + 1:
       X = X + 1:
       GOSUB 11000:
       GOTO 3060
3180   PRINT BP$:
       FS = 0:
       GOTO 3100
3190   IF CC > 1 THEN SO =  - 1:
       CC = CC - 1:
       X = X - 1:
       GOSUB 11000:
       GOTO 3060
3200   PRINT BP$;:
       FS = 0:
       GOTO 3040
3210   IF FS > 8 THEN FS = 0:
       GOTO 3060
3220   IF FS = 8 THEN FS = 0:
       GOTO 3100
3230   RETURN
3240   REM  ------------------
3250   REM  LECTURA DE UN CAMPO
3260   REM  ------------------
3270   FS = 0
3280   IF TP(NC) = 4 THEN  RETURN
```

```
3290  IF TP(NC) = 2 THEN B$(NC) =
       LEFT$ (B$(NC),LC(NC)):
      CC =  LEN (B$(NC)) + 1:
      GOSUB 4080:
      RETURN
3300  IF  MID$ (B$(NC),CC,1) = PD$
       THEN X = X + SO:
      CC = CC + SO
3310  GOSUB 11000
3320  GET A$:
      IF A$ = "" THEN 3320
3330  CA =  ASC (A$):
      IF CA > 31 THEN 3380
3340  FOR I = 1 TO 15
3350     IF TF(I) < > (CA) THEN 3360
3355     FS = I:
         GOSUB 3630:
         IF EE = 1 THEN EE = 0:
         GOTO 3070
3357     RETURN
3360  NEXT I
3370  GOTO 3320
3380  IF TP(NC) < > 1 THEN 3440
3390  IF CA = 45 OR CA = 46 OR (CA
       > = 48 AND CA < = 57) THEN 341
      0
3400  PRINT BP$;:
      GOTO 3320
3410  IF CA < > 46 THEN 3440
3420  GOSUB 4270
3430  IF SWP = 1 THEN 3400
3440  SO = 1:
      GOSUB 3520
3450  IF CC = LC(NC) THEN  RETURN
3460  CC = CC + 1:
      X = X + 1:
      GOTO 3300
3520  REM -----------------------
3530  REM  TRANSFERENCIA AL BUFFER
3540  REM -----------------------
3550  IF CC = 1 THEN SN$ = "":
      GOTO 3570
3560  SN$ =  LEFT$ (B$(NC),CC - 1)
3570  IF CC = LC(NC) THEN DS$ = "":
      GOTO 3590
3580  N = LC(NC) - CC:
      DS$ =  RIGHT$ (B$(NC),N)
3590  B$(NC) = SN$ + A$ + DS$
3620  PRINT A$:
      RETURN
3630  REM --------
3640  REM  COMANDOS
3650  REM --------
3660  IF FS > = 5 THEN 3690
3670  IF FS = 5 AND TP(NC) = 1 THEN
       GOSUB 4080
3680  RETURN
3690  IF FS = 9 THEN  GOSUB 13000:
      RETURN
3700  IF FS = 8 THEN  GOSUB 4080:
      RETURN
3710  IF FS < 8 THEN  RETURN
3720  ON FS - 9 GOSUB 3740,3910
3730  RETURN
3740  REM ---------
3750  REM  INSERCION
3760  REM ---------
3770  IF CC = LC(NC) THEN  PRINT BP$:
      RETURN
3780  IF CC = 1 THEN SN$ = "":
      GOTO 3800
3790  SN$ =  LEFT$ (B$(NC),CC - 1)
3800  N = LC(NC) - CC + 1:
      DS$ =  RIGHT$ (B$(NC),N)
3820  C$ = BL$
3830  A$ = SN$ + C$ + DS$
3840  B$(NC) =  LEFT$ (A$,LC(NC))
3850  XX = X:
      YY = Y
3860  X = CX(NC):
      Y = CY(NC):
      GOSUB 11000
3870  X = XX:
      Y = YY
3880  PRINT B$(NC):
      SO =  - 1
3890  IF  RIGHT$ (B$(NC),1) = PD$
       THEN 3780
3900  RETURN
3910  REM -------
3920  REM  BORRADO
3930  REM -------
3940  IF TP(NC) < > 1 THEN 3960
3950  IF  MID$ (B$(NC),2,1) = PD$
       AND CC = 1 THEN  RETURN
3960  IF CC < 1 THEN  PRINT BP$:
      RETURN
3970  IF CC = LC(NC) THEN DS$ = "":
      GOTO 4000
3980  N = LC(NC) - CC:
      DS$ =  RIGHT$ (B$(NC),N)
3990  IF CC = 1 THEN SN$ = "":
      GOTO 4010
4000  SN$ =  LEFT$ (B$(NC),CC - 1)
4010  IF TP(NC) = 1 OR TP(NC) = 2
       THEN C$ = NS$:
      GOTO 4030
4020  C$ = AS$
4030  B$(NC) = SN$ + DS$ + C$
4040  XX = X:
      YY = Y
4050  X = CX(NC):
      Y = CY(NC):
      GOSUB 11000
4060  X = XX:
      Y = YY
4070  PRINT B$(NC):
      SO = 1:
      RETURN
4080  REM ---------
4090  REM  ALINEADOS
4100  REM ---------
4110  IF LC(NC) = 1 THEN  RETURN
4120  N1 = LC(NC) - CC + 1
4130  IF CC = 1 THEN B$(NC) = "":
      GOTO 4150
4140  B$(NC) =  LEFT$ (B$(NC),CC
       - 1)
4150  IF TP(NC) = 1 OR TP(NC) = 2
       THEN 4200
4160  FOR I = 1 TO N1
4170     B$(NC) = B$(NC) + BL$
4180  NEXT I
4190  GOTO 4240
4200  FOR I = 1 TO N1
4210     B$(NC) = BL$ + B$(NC)
4220  NEXT I
4230  IF  RIGHT$ (B$(NC),1) = PD$
       THEN B$(NC) = BL$ +  LEFT$ (B$(N
      C),LC(NC) - 1)
4240  X = CX(NC):
      Y = CY(NC)
4250  GOSUB 11000
4260  PRINT B$(NC):
      RETURN
4270  REM ----------
4280  REM  CTRL.PUNTO
4290  REM ----------
4300  SWP = 0
4310  IF CC = 1 OR CC = LC(NC) THEN SWP
       = 1:
      RETURN
4320  FOR I = 1 TO LC(NC)
4330     IF  MID$ (B$(NC),I,1) = PD$
          THEN I = LC(NC):
         SWP = 1:
         RETURN
4340  NEXT I:
      RETURN
5000  REM ------------------
5010  REM  INTRODUCCION SIGLA
5020  REM ------------------
5030  EE = 0:
      HOME :
      YT = 7:
      TI$ = "INTRODUCIR LA SIGLA DEL TE
      ST.":
      GOSUB 11200
5040  X = 16:
      Y = 13:
      GOSUB 11000:
      PRINT "SIGLA "; CHR$ (95);
       CHR$ (95)
5050  X = 21:
      GOSUB 11000:
      INPUT S$:
      IF S$ = "" THEN 5040
5060  S$ =  LEFT$ (S$,2)
5070  NF$ = "AT" + S$:
      LR = 30:
      GOSUB 6000
5080  E = 3:
      NR = 1:
      GOSUB 6200
5090  IF EE = 0 THEN E = 4:
      GOSUB 5500
5100  IF EE = 1 THEN  GOSUB 6600:
      GOSUB 6000:
      GOTO 220
5110  RETURN
5500  REM -------------
5510  REM  ACTUALIZACION
5520  REM -------------
5530  FOR I = 1 TO 10:
         NR = I:
         GOSUB 6200
5540     B$(I) = AA$:
      NEXT I
5550  RETURN
6000  REM ----------------
6010  REM  APERTURA FICHERO
6020  REM ----------------
6030  PRINT D$
6040  PRINT D$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D$
6060  RETURN
6200  REM ----------------
6210  REM  LECTURA REGISTRO
6220  REM ----------------
6230  PRINT D$
6240  PRINT D$;"READ";NF$;",R";NR
6250  INPUT AA$
6260  PRINT D$
6270  RETURN
6400  REM ------------------
6410  REM  ESCRITURA REGISTRO
6420  REM ------------------
6430  PRINT D$
6440  PRINT D$;"WRITE";NF$;",R";NR
6450  PRINT  CHR$ (34) + AA$
6460  PRINT D$
6470  RETURN
6600  REM --------------
6610  REM  CIERRA FICHERO
6620  REM --------------
6630  PRINT D$
6640  PRINT D$;"CLOSE";NF$
6650  PRINT D$
6660  RETURN
8000  REM -----------------------
8010  REM  INTRODUCCION ATRIBUTOS
8020  REM -----------------------
8030  NUC = 10:
      FOR I = 1 TO 10:
         TP(I) = E
8040     LC(I) = 30:
```

```
        ND(I) = 0:
        DX(I) = 4
 8050   CX(I) = 7:
        DY(I) = I + 9:
        CY(I) = I + 9
 8060   D$(I) =  STR$ (I) + ".":
     NEXT I
 8070 IN$ = "INTRODUCIR LOS ATRIBUTOS."
     :
     CO = 0
 8080 GOSUB 2000:
     IF E = 4 THEN  FOR I = 1 TO 10:
       TP(I) = 3:
     NEXT I
 8090 GOSUB 3000:
     IF FS = 6 THEN 8080
 8100 IF FS = 7 THEN  RETURN
 8110 FOR I = 1 TO 10:
       NR = I:
       AA$ = B$(I)
 8120   GOSUB 6400:
     NEXT I
 8130 GOSUB 6600:
     RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA 23,26,19,1,5,27,18,13,25,9,4
     ,0,0,0,0
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 HTAB X:
     VTAB Y
11040 RETURN
11100 REM ---------------
11110 REM ESPERA CARACTER
11120 REM ---------------
11130 GOSUB 11600:
     YT = 23:
     TI$ = "PULSAR UNA TECLA":
     FLASH :
     GOSUB 11200
11140 PRINT BF$;:
     X = 20:
     Y = 23:
     GOSUB 11000:
     GET A$:
     GOSUB 11600
11150 RETURN
11200 REM ---------------
11210 REM CENTRA EN VIDEO
11220 REM ---------------
11230 IF TI$ = "" THEN  RETURN
11235 TI$ =  LEFT$ (TI$,40)
11240 XT = (42 -  LEN (TI$)) / 2
11250 HTAB XT:
     VTAB YT
11260 PRINT TI$;:
     RETURN
11600 REM ------------------
11610 REM BORRA ULTIMA LINEA
11620 REM ------------------
11630 Y = 23:
     X = 1:
     GOSUB 11000
11640 NORMAL :
     PRINT  LEFT$ (BA$,40);:
     Y = 10:
     GOSUB 11000
11650 RETURN
12000 REM ----------------------
12010 REM INTRODUCCION ELEMENTOS
12020 REM ----------------------
12030 NF$ = "EL" + S$:
     LR = 30:
     GOSUB 6000
12032 IF E = 4 THEN NR = 1:
     GOSUB 6200:
     MX =  VAL (AA$)
12035 IR = 1:
     JR = 1
12040 E1 = 1:
     IF E = 4 THEN E1 = 4:
     GOSUB 12500:
     GOTO 12050
12045 NR = 1:
     AA$ =  STR$ (IR):
     GOSUB 6400
12050 NUC = 11:
     TP(1) = E:
     LC(1) = 20
12060 DX(1) = 3:
     DY(1) = 9:
     CX(1) = 13:
     CY(1) = 9
12070 D$(1) = "ELEMENTO:
     ":
     FOR I = 2 TO 11
12080   TP(I) = E1:
        LC(I) = 1:
        DX(I) = 15
12090   DY(I) = I + 12:
        CX(I) = 18:
        CY(I) = I + 12:
        D$(I) =  STR$ (I - 1)
12100 NEXT I:
     IN$ = "INTRODUCIR ELEMENTOS E IND
     ICADORES"
12110 GOSUB 2000:
     IF E = 4 THEN TP(1) = 3:
     FOR I = 2 TO 11:
       TP(I) = 1:
     NEXT I
12120 GOSUB 3000:
     IF FS = 6 THEN 12110
12130 IF FS = 7 THEN  RETURN
12140 IR = IR + 1:
     NR = IR:
     AA$ = B$(1):
     FOR I = 2 TO 11:
       AA$ = AA$ +  LEFT$ (B$(I),1):
     NEXT I
12150 GOSUB 6400
12160 HOME :
     YT = 13:
     TI$ = "OTRO ELEMENTO ? (S/N)":
     GOSUB 11200
12170 GET A$:
     IF A$ <  > "S" AND A$ <  > "N"
      THEN 12160
12180 IF A$ = "S" THEN  GOTO 12040
12190 NR = 1:
     AA$ =  STR$ (IR):
     GOSUB 6400
12200 GOSUB 6600:
     RETURN
12500 REM -------------
12510 REM ACTUALIZACION
12520 REM -------------
12530 NR = IR + 1:
     IF NR > MX THEN AA$ = "":
     GOTO 12540
12535 GOSUB 6200
12540 B$(1) =  LEFT$ (AA$,20)
12550 FOR I = 2 TO 11:
       B$(I) =  MID$ (AA$,I + 19,1)
12560 NEXT I:
     RETURN
13000 REM -----
13010 REM AYUDA
13020 REM -----
13030 XX = X:
     YY = Y
13040 INVERSE
13050 K = 1
13060 X = 1:
     Y = 5:
     GOSUB 11000
13070 PRINT HE$(K);
13080 B =  PEEK ( - 16384)
13090 POKE  - 16368,0
13100 IF B < 128 THEN 13080
13110 IF B <  > 160 THEN 13080
13120 K = K + 1:
     IF K > 8 THEN 13140
13130 GOTO 13060
13140 Y = 5:
     X = 1:
     GOSUB 11000:
     PRINT  LEFT$ (BA$,40);
13150 YT = 5:
     TI$ = HE$:
     GOSUB 11200
13160 NORMAL
13170 X = XX:
     Y = YY
13180 RETURN
40000 REM -------------
40010 REM GESTION ERROR
40020 REM -------------
40030 IF  PEEK (222) = 5 THEN 40130
40040 IF  PEEK (222) <  > 8 THEN 40060
40050 TEXT :
     HOME :
     YT = 13:
     TI$ = "COMPROBAR EL DISCO DATOS":
     INVERSE :
     GOSUB 11200:
     PRINT :
     END
40060 IF  PEEK (222) <  > 13 AND
      PEEK (222) <  > 11 THEN 40080
40070 TEXT :
     HOME :
     YT = 13:
     TI$ = "NOMBRE FICHERO NO VALIDO":
     INVERSE :
     GOSUB 11200:
     PRINT :
     END
40080 IF  PEEK (222) <  > 9 THEN 40100
40090 TEXT :
     HOME :
     YT = 13:
     TI$ = "DISCO LLENO":
     INVERSE :
     GOSUB 11200:
     PRINT :
     END
40100 IF  PEEK (222) <  > 4 THEN 40115
40110 TEXT :
     HOME :
     YT = 13:
     TI$ = "DISCO PROTEGIDO":
     INVERSE :
     GOSUB 11200:
     PRINT :
     END
40115 L =  PEEK (218) + 256 *  PEEK (21
     9)
40120 TEXT :
     HOME :
     YT = 13:
     TI$ = "ERROR " +  STR$ ( PEEK (22
     2)) + " EN " +  STR$ (L):
     GOSUB 11200:
     END
40130 IF E = 3 THEN EE = 1:
     GOTO 5090
40140 TI$ = "LOS DATOS NO EXISTEN":
     INVERSE :
     YT = 13:
     GOSUB 11200
40150 END
```

Atributos-Uso

```
10    REM   ------------------
20    REM   TEST-ATRIBUTOS-USO
30    REM      VERSION  DOS
40    REM   ------------------
60    DIM TF(15),TP(20),LC(20),ND(20),C
      X(20),CY(20),DX(20),DY(20),B$(20)
      ,D$(20)
70    BP$ =  CHR$ (7):
      NS$ = "#":
      AS$ =  CHR$ (95):
      PD$ = ".":
      BL$ = " ":
      D$ =  CHR$ (4):
      US$ =  CHR$ (13):
      ZE$ = "0":
      GIU$ = " ":
      SL$ = "/"
80    B$ = "         ":
      BA$ = B$ + B$ + B$ + B$ + B$:
      B$ = ""
90    HE$ = "AYUDA CTRL - Y"
100   HE$(1) = "CTRL Y. AYUDA ('ESPACIO
      ' PARA CONTINUAR)"
110   HE$(2) = "ESC. ANULA TODA LA INTR
      ODUCCION         "
120   HE$(3) = "RETURN. ALINEA Y CONVAL
      IDA INTRODUCCION "
130   HE$(4) = "CTRL I. PERMITE LA INSE
      RCION DE ESPACIOS"
140   HE$(5) = "CTRL D. BORRA UN CARACT
      ER               "
150   HE$(6) = "CTRL R. VUELVE AL MENU
      PRINCIPAL        "
160   HE$(7) = "CTRL A,W,S,Z.MUEVEN EN
      LAS 4 DIRECCIONES"
170   HE$(8) = "CTRL E. INTRODUCCION EN
       DISCO           "
172   TEXT :
      HOME :
      YT = 12:
      TI$ = "CUANTAS UNIDADES HAY ? (1/
      2)":
      GOSUB 11200
174   GET A$:
      IF A$ < "1" OR A$ > "2" THEN 172
176   DR$ = A$:
      PRINT DR$
178   HOME :
      YT = 12:
      TI$ = "INTRODUCIR DISCO DATOS EN
      LA UNIDAD " + DR$:
      GOSUB 11200:
      GOSUB 11100
179   ONERR  GOTO 40000
180   MP$ = "TEST - ATRIBUTOS - USO"
190   RESTORE :
      FOR I = 1 TO 10:
         READ TF(I):
      NEXT I
195   DIM NO$(100),FO$(100)
220   GOSUB 500
230   FS = 0:
      GOSUB 5000
240   GOSUB 8000
245   IF FS = 7 THEN 230
250   GOSUB 12000
260   IF FS = 7 THEN 230
410   Y = 5:
      X = 1:
      GOSUB 11000:
      INVERSE :
      PRINT  LEFT$ (BA$,40);:
      NORMAL
460   GOTO 230
500   REM   -----------------
510   REM   PRESENTA CABECERA
520   REM   -----------------
530   POKE 34,0:
      HOME :
      INVERSE :
      PRINT " E. G. S. " +  LEFT$ (BA$
      ,7) + "EDICIONES FORUM, S.A."
       + "  ";:
      NORMAL
540   YT = 3:
      TI$ = MP$:
      GOSUB 11200
550   Y = 2:
      X = 1:
      INVERSE :
      GOSUB 11000:
      PRINT " ":
      X = 40:
      GOSUB 11000:
      PRINT " ";:
      Y = 3:
      X = 1:
      GOSUB 11000:
      PRINT " ":
      X = 40:
      GOSUB 11000:
      PRINT " ";:
      Y = 4:
      X = 1:
      GOSUB 11000:
      PRINT " ":
      X = 40:
      GOSUB 11000:
      PRINT " ";:
      Y = 5:
      X = 1:
      GOSUB 11000
560   PRINT  LEFT$ (BA$,40);:
      NORMAL :
      POKE 34,5
570   RETURN
2000  REM  ----------------
2010  REM  PRESENTA MASCARA
2020  REM  ----------------
2030  HOME
2035  YT = 5:
      TI$ = HE$:
      INVERSE :
      GOSUB 11200:
      NORMAL
2040  YT = 7:
      TI$ = IN$:
      GOSUB 11200
2050  IF CO <  > 0 THEN Y = 7:
      X = XT +  LEN (IN$) + 1:
      GOSUB 11000:
      PRINT CO
2060  SP = 33 -  LEN (VO$(OPZ))
2070  SP$ = ""
2080  IF SP = 0 THEN 2100
2090  FOR J = 1 TO SP:
         SP$ = SP$ + BL$:
      NEXT
2100  LP =  LEN (PROG$)
2110  SI =  INT ((40 - LP) / 2):
      DE = 40 - LP - SI
2120  SI$ = "":
      DE$ = ""
2130  FOR J = 1 TO SI
2140     SI$ = SI$ + BL$
2150  NEXT
2160  FOR J = 1 TO DE
2170     DE$ = DE$ + BL$
2180  NEXT
2240  REM  MASCARA
2250  FOR J = 1 TO NUC
2260     A$ = ""
2270     X = DX(J):
         Y = DY(J):
         GOSUB 11000:
         :
         PRINT D$(J)
2280     NORMAL
2290     IF TP(J) <  > 5 AND TP(J)
          <  > 4 THEN 2330
2300     X = CX(J):
         Y = CY(J):
         GOSUB 11000
2310     PRINT B$(J)
2320     GOTO 2500
2330     N1 = LC(J):
         N2 = ND(J):
         IF TP(J) = 1 OR TP(J) = 2
          THEN C$ = NS$:
         GOTO 2350
2340     C$ = AS$
2345     IF N1 = 1 THEN A$ = C$:
         GOTO 2390
2350     FOR I = 1 TO N1 - N2 - 1:
            A$ = A$ + C$:
         NEXT I
2360     IF N2 = 0 THEN A$ = A$ + C$:
         GOTO 2390
2370     A$ = A$ + PD$
2380     FOR I = 1 TO N2:
            A$ = A$ + ZE$:
         NEXT I
2390     X = CX(J):
         Y = CY(J):
         GOSUB 11000
2400     PRINT A$
2410     B$(J) = ""
2420     FOR K = 1 TO N1 - N2 - 1
2430        B$(J) = B$(J) + BL$
2440     NEXT K
2450     IF N2 = 0 THEN B$(J) = B$(J)
          + BL$:
         GOTO 2500
2460     B$(J) = B$(J) + PD$
2470     FOR K = 1 TO N2
2480        B$(J) = B$(J) + BL$
2490     NEXT K
2500  NEXT J
2510  RETURN
2600  REM  ----------------
2610  REM  COMPLETA MASCARA
2620  REM  ----------------
2630  YT = 9:
      TI$ = "'S' PARA 'SI' Y 'N' PARA '
      NO'":
      GOSUB 11200
2640  RETURN
3000  REM  ----------
3010  REM  DATA ENTRY
3020  REM  ----------
3030  J = 1
3040  NC = J:
      X = CX(NC):
      Y = CY(NC)
3050  CC = 1
3060  GOSUB 3240
3070  IF FS = 0 THEN 3100
3080  IF FS > 4 THEN 3210
3090  GOTO 3120
3100  J = J + 1:
      IF J > NUC THEN 3030
3110  GOTO 3040
3120  ON FS GOTO 3130,3150,3170,3190
3130  IF J > 1 THEN J = J - 1:
      GOTO 3040
3140  PRINT BP$;:
      FS = 0:
      J = NUC:
      GOTO 3040
3150  IF J < NUC THEN J = J + 1:
      GOTO 3040
3160  PRINT BP$;:
      FS = 0:
      GOTO 3030
3170  IF CC < LC(J) THEN SO = 1:
```

```
      CC = CC + 1:
      X = X + 1:
      GOSUB 11000:
      GOTO 3060
3180  PRINT BP$:
      FS = 0:
      GOTO 3100
3190  IF CC > 1 THEN SO =  - 1:
      CC = CC - 1:
      X = X - 1:
      GOSUB 11000:
      GOTO 3060
3200  PRINT BP$;:
      FS = 0:
      GOTO 3040
3210  IF FS > 8 THEN FS = 0:
      GOTO 3060
3220  IF FS = 8 THEN FS = 0:
      GOTO 3100
3230  RETURN
3240  REM  ------------------
3250  REM  LECTURA DE UN CAMPO
3260  REM  ------------------
3270  FS = 0
3280  IF TP(NC) = 4 THEN  RETURN
3290  IF TP(NC) = 2 THEN B$(NC) =
       LEFT$ (B$(NC),LC(NC)):
      CC =  LEN (B$(NC)) + 1:
      GOSUB 4080:
      RETURN
3300  IF  MID$ (B$(NC),CC,1) = PD$
       THEN X = X + SO:
      CC = CC + SO
3310  GOSUB 11000
3320  GET A$:
      IF A$ = "" THEN 3320
3330  CA =  ASC (A$):
      IF CA > 31 THEN 3380
3340  FOR I = 1 TO 15
3350     IF TF(I) = (CA) THEN FS
         = I:
         GOSUB 3630:
         RETURN
3360  NEXT I
3370  GOTO 3320
3380  IF TP(NC) <  > 1 THEN 3440
3390  IF CA = 45 OR CA = 46 OR (CA
       >  = 48 AND CA <  = 57) THEN 341
      0
3400  PRINT BP$;:
      GOTO 3320
3410  IF CA <  > 46 THEN 3440
3420  GOSUB 4270
3430  IF SWP = 1 THEN 3400
3440  SO = 1:
      GOSUB 3520
3450  IF CC = LC(NC) THEN  RETURN
3460  CC = CC + 1:
      X = X + 1:
      GOTO 3300
3520  REM  ----------------------
3530  REM  TRANSFERENCIA AL BUFFER
3540  REM  ----------------------
3550  IF CC = 1 THEN SN$ = "":
      GOTO 3570
3560  SN$ =  LEFT$ (B$(NC),CC - 1)
3570  IF CC = LC(NC) THEN DS$ = "":
      GOTO 3590
3580  N = LC(NC) - CC:
      DS$ =  RIGHT$ (B$(NC),N)
3590  B$(NC) = SN$ + A$ + DS$
3620  PRINT A$:
      RETURN
3630  REM  --------
3640  REM  COMANDOS
3650  REM  --------
3660  IF FS >  = 5 THEN 3690
3670  IF FS = 5 AND TP(NC) = 1 THEN
       GOSUB 4080
3680  RETURN
3690  IF FS = 9 THEN  GOSUB 13000:
      RETURN
3700  IF FS = 8 THEN  GOSUB 4080:
      RETURN
3710  IF FS < 8 THEN  RETURN
3720  ON FS - 9 GOSUB 3740,3910
3730  RETURN
3740  RETURN
3910  RETURN
4080  RETURN
4270  RETURN
5000  REM  ---------------
5010  REM  INTRODUCCION SIGLA
5020  REM  ---------------
5030  HOME :
      YT = 7:
      TI$ = "INTRODUCIR LA SIGLA DEL TE
      ST:
      ":
      GOSUB 11200
5040  X = 17:
      Y = 12:
      GOSUB 11000
5050  PRINT "SIGLA "; CHR$ (95);
       CHR$ (95)
5060  X = 22:
      GOSUB 11000:
      INPUT S$:
      IF S$ = "" THEN 5040
5070  NF$ = "AT" + S$:
      LR = 30:
      GOSUB 6000
5080  FOR NR = 1 TO 10:
         GOSUB 6200
5090     DT$(NR) = AA$:
      NEXT NR
5100  HOME :
      YT = 13:
      TI$ = "DESEAS VER LOS ATRIBUTOS ?
       (S/N)":
      GOSUB 11200
5110  GET A$:
      IF A$ <  > "S" AND A$ <  > "N"
       THEN 5100
5120  IF A$ = "N" THEN  GOSUB 6600:
      RETURN
5130  HOME :
      X = 3:
      FOR Y = 10 TO 19:
         GOSUB 11000
5140     PRINT Y - 9;". ";DT$(Y - 9)
5150  NEXT Y:
      GOSUB 11100:
      GOSUB 6600:
      RETURN
6000  REM  ---------------
6010  REM  APERTURA FICHERO
6020  REM  ---------------
6030  PRINT D$
6040  PRINT D$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D$
6060  RETURN
6200  REM  ---------------
6210  REM  LECTURA REGISTRO
6220  REM  ---------------
6230  PRINT D$
6240  PRINT D$;"READ";NF$;",R";NR
6250  INPUT AA$
6260  PRINT D$
6270  RETURN
6600  REM  -------------
6610  REM  CIERRA FICHERO
6620  REM  -------------
6630  PRINT D$
6640  PRINT D$;"CLOSE";NF$
6650  PRINT D$
6660  RETURN
8000  REM  ------------------
8010  REM  CARGA LOS ELEMENTOS
8020  REM  ------------------
8030  NF$ = "EL" + S$:
      LR = 30:
      GOSUB 6000
8040  NR = 1:
      GOSUB 6200:
      MX =  VAL (AA$)
8050  FOR I = 2 TO MX:
         NR = I:
         GOSUB 6200
8060     NO$(I - 1) =  LEFT$ (AA$,20)
8070     FO$(I - 1) =  MID$ (AA$,21,10)

8080  NEXT I:
      GOSUB 6600:
      RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA 23,26,19,1,5,27,18,13,25,9,4
      ,0,0,0,0
11000 REM ---------------
11010 REM POSICIONA CURSOR
11020 REM ---------------
11030 HTAB X:
      VTAB Y
11040 RETURN
11100 REM --------------
11110 REM ESPERA CARACTER
11120 REM --------------
11130 GOSUB 11600:
      YT = 23:
      TI$ = "PULSAR UNA TECLA":
      FLASH :
      GOSUB 11200
11140 PRINT BP$;:
      X = 20:
      Y = 23:
      GOSUB 11000:
      GET A$:
      GOSUB 11600
11150 RETURN
11200 REM --------------
11210 REM CENTRA EN VIDEO
11220 REM --------------
11230 IF TI$ = "" THEN  RETURN
11235 TI$ =  LEFT$ (TI$,40)
11240 XT = (42 -  LEN (TI$)) / 2
11250 HTAB XT:
      VTAB YT
11260 PRINT TI$;:
      RETURN
11600 REM -----------------
11610 REM BORRA ULTIMA LINEA
11620 REM -----------------
11630 Y = 23:
      X = 1:
      GOSUB 11000
11640 NORMAL :
      PRINT  LEFT$ (BA$,40);:
      Y = 10:
      GOSUB 11000
11650 RETURN
12000 REM ------------------
12010 REM DESARROLLO DEL TEST
12020 REM ------------------
12030 HOME :
      YT = 13
12040 TI$ = "DESARROLLO, LINIA O COLUMN
      A ? (L/C)"
12050 GOSUB 11200:
      GET A$:
      IF A$ <  > "L" AND A$ <  > "C"
       THEN 12040
12060 IF A$ = "C" THEN 12300
12070 N9 =  RND (1) * (MX - 1) + 1
12080 IF  LEFT$ (NO$(N9),1) = " "
```

```
      THEN 12070
12090 D$(1) = "ELEMENTO:
      ":
      B$(1) = NO$(N9)
12100 TP(1) = 4:
      LC(1) = 20:
      DX(1) = 1:
      DY(1) = 12
12110 CX(1) = 11:
      CY(1) = 12
12120 FOR I = 2 TO 11:
         D$(I) = DT$(I - 1)
12130    LC(I) = 1:
         TP(I) = 3:
         DX(I) = 4
12140    DY(I) = 12 + I:
         CX(I) = 27:
         CY(I) = 12 + I
12150 NEXT I:
      NUC = 11
12160 IN$ = "INDICAR LOS ATRIBUTOS CORR
      ECTOS":
      CO = 0
12165 FOR I = 2 TO 11:
         B$(I) = "":
      NEXT I
12170 GOSUB 2000:
      GOSUB 2600
12180 GOSUB 3000:
      IF FS = 6 THEN 12170
12190 IF FS = 7 THEN  RETURN
12200 R$ = "":
      FOR I = 2 TO 11
12210    IF B$(I) = "S" THEN R$ = R$
          + "1"
12220    IF B$(I) = "N" THEN R$ = R$
          + "0"
12230 NEXT I
12240 L =  LEN (R$):
      IF L = 0 THEN R$ = "EE":
      L = 1
12242 IF D$(L + 2) <  >  LEFT$ (BA$,30)
       AND D$(L + 2) <  > "" THEN R$
       = "EE"
12245 RO$ =  LEFT$ (FO$(N9),L)
12250 IF R$ <  > RO$ THEN  HOME :
      YT = 13:
      TI$ = "HAY UN ERROR. COMPRUEBA":
      INVERSE :
      GOSUB 11200:
      PRINT BP$;:
      GOSUB 11100:
      GOTO 12160
12255 HOME :
      YT = 13:
      TI$ = "MUY BIEN, TEST SUPERADO":
      INVERSE :
      GOSUB 11200:
      GOSUB 11100
12260 HOME :
      YT = 13:
      TI$ = "OTRA VEZ ?(S/N)":
      GOSUB 11200
12270 GET A$:
      IF A$ <  > "S" AND A$ <  > "N"
       THEN 12260
12280 IF A$ = "S" THEN 12030
12290 FS = 7:
      RETURN
12300 N9 =  RND (1) * 10 + 1
12310 B$(1) = DT$(N9):
      K = 1:
      RO$ = "":
      IF B$(1) =  LEFT$ (BA$,30)
       THEN 12300
12320 N8 =  RND (1) * (MX - 1) + 1
12330 IF NO$(N8) =  LEFT$ (BA$,20)
       THEN 12320
12340 D$(K + 1) = NO$(N8):
      RO$ = RO$ +  MID$ (FO$(N8),N9,1)
12350 IF K < 10 THEN K = K + 1:
      GOTO 12320
12360 IN$ = "INDICAR LOS ELEMENTOS CORR
      ECTOS":
      CO = 0
12370 D$(1) = "ATRIB.:
      ":
      TP(1) = 4:
      LC(1) = 30
12380 DX(1) = 1:
      DY(1) = 12:
      CX(1) = 10:
      CY(1) = 12
12390 FOR I = 2 TO 11:
         TP(I) = 3:
         LC(I) = 1
12400    DX(I) = 9:
         DY(I) = 12 + I:
         CX(I) = 32:
         CY(I) = 12 + I
12410 NEXT I:
      NUC = 11
12420 GOSUB 2000:
      GOSUB 2600
12430 GOSUB 3000:
      IF FS = 6 THEN 12420
12440 IF FS = 7 THEN  RETURN
12450 R$ = "":
      FOR I = 1 TO 10
12460    IF B$(I + 1) = "S" THEN R$
          = R$ + "1"
12470    IF B$(I + 1) = "N" THEN R$
          = R$ + "0"
12480 NEXT I
12490 IF RO$ <  > R$ THEN  HOME :
      YT = 13:
      TI$ = "HAY UN ERROR COMPRUEBA":
      INVERSE :
      GOSUB 11200:
      PRINT BP$;:
      GOSUB 11100:
      GOTO 12420
12500 HOME :
      YT = 13:
      TI$ = "MUY BIEN, TEST SUPERADO":
      INVERSE :
      GOSUB 11200:
      GOSUB 11100
12510 YT = 13:
      HOME :
      TI$ = "OTRA VEZ ?(S/N)":
      GOSUB 11200
12520 GET A$:
      IF A$ <  > "S" AND A$ <  > "N"
       THEN 12510
12530 IF A$ = "S" THEN 12030
12540 FS = 7:
      RETURN
13000 REM -----
13010 REM AYUDA
13020 REM -----
13030 XX = X:
      YY = Y
13040 INVERSE
13050 K = 1
13060 X = 1:
      Y = 5:
      GOSUB 11000
13070 PRINT HE$(K);
13080 B =  PEEK ( - 16384)
13090 POKE  - 16368,0
13100 IF B < 128 THEN 13080
13110 IF B <  > 160 THEN 13080
13120 K = K + 1:
      IF K > 8 THEN 13140
13130 GOTO 13060
13140 Y = 5:
      X = 1:
      GOSUB 11000:
      PRINT  LEFT$ (BA$,40);
13150 YT = 5:
      TI$ = HE$:
      GOSUB 11200
13160 NORMAL
13170 X = XX:
      Y = YY
13180 RETURN
40000 REM -------------
40010 REM GESTION ERROR
40020 REM -------------
40030 IF  PEEK (222) = 5 THEN 40130
40040 IF  PEEK (222) <  > 8 THEN 40115
40050 HOME :
      YT = 13:
      TI$ = "COMPROBAR EL DISCO DATOS":
      INVERSE :
      GOSUB 11200:
      NORMAL :
      END
40115 L =  PEEK (218) + 256 *  PEEK (21
      9)
40120 HOME :
      YT = 13:
      TI$ = "ERROR " +  STR$ ( PEEK (22
      2)) + " EN " +  STR$ (L):
      GOSUB 11200:
      END
40130 HOME :
      TI$ = "EL TEST NO EXISTE":
      YT = 13:
      INVERSE :
      GOSUB 11200:
      GOSUB 11100:
      GOTO 220
```

Símbolos-Generación

```
10    REM  ------------------------
20    REM   TEST-GENERACION SIMBOLOS
30    REM        VERSION   DOS
40    REM  ------------------------
50    D$ =  CHR$ (4):
      BP$ =  CHR$ (7)
60    B$ = "          ":
      BA$ = B$ + B$ + B$ + B$:
      B$ = ""
70    PRINT D$:
      PRINT D$;"BLOAD CAR,D1":
      PRINT D$
80    HGR2 :
      X = 8:
      Y = 12:
      A$ = "CUANTAS UNIDADES HAY ? (1/2
      )"
90    GOSUB 20300:
      GET C$:
      IF C$ <  > "1" AND C$ <  > "2"
       THEN 80
100   X = 2:
      A$ = "INTRODUCIR DISCO DATOS EN L
      A UNIDAD " + C$:
      GT = 1:
      GOSUB 20300
105   X = 12:
      Y = 23:
      A$ = "PULSAR UNA TECLA":
      GT = 0:
      GOSUB 20300:
      GET A$
110   DR$ = C$
190   GOSUB 12000:
      TEXT :
      HOME :
      END
6000  REM  ----------------
6010  REM  APERTURA FICHERO
```

```
6020  REM  -----------------
6030  PRINT D$
6040  PRINT D$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D$
6060  RETURN
6400  REM  ------------------
6410  REM  ESCRITURA REGISTRO
6420  REM  ------------------
6430  PRINT D$
6440  PRINT D$;"WRITE";NF$;",R";NR
6450  PRINT  CHR$ (34) + AA$
6460  PRINT D$
6470  RETURN
6600  REM  -------------
6610  REM  CIERRA FICHERO
6620  REM  -------------
6630  PRINT D$
6640  PRINT D$;"CLOSE";NF$
6650  PRINT D$
6660  RETURN
12000 REM -----
12010 REM START
12020 REM -----
12030 NF$ = "SIMBOLOS":
      LR = 75:
      GOSUB 6000
12050 IR = 1:
      TE = 1
12060 GOSUB 35000:
      GOSUB 35500
12062 X = 2:
      Y = 19:
      A$ = "INDICAR LA RESPUESTA CORREC
      TA (3,4,5)":
      GOSUB 20300
12064 GET C$:
      IF C$ < "3" OR C$ > "5" THEN 1206
      2
12065 NR = NR + 1:
      AA$ = C$:
      GOSUB 6400
12067 A$ =  LEFT$ (BA$,37):
      X = 2:
      GOSUB 20300
12070 A$ = "OTRO TEST ? (S/N)":
      X = 11:
      Y = 19
12080 GOSUB 20300:
      GET C$:
      IF C$ <  > "S" AND C$ <  > "N"
       THEN 12070
12090 IF C$ = "S" THEN IR = IR + 6:
      TE = TE + 1:
      GOTO 12060
12100 NR = 1:
      AA$ =  STR$ (TE):
      GOSUB 6400
12110 GOSUB 6600:
      RETURN
20000 REM -----------------------
20010 REM ESPERA Y PARPADEO CURSOR
20020 REM -----------------------
20030 A0 =  PEEK ( - 16384):
      IF A0 > 128 THEN  POKE  - 16368,0
      :
      A0 = A0 - 128:
      RETURN
20040 T = T + 1:
      IF T < 10 THEN 20030
20050 T = 1:
      IF  PEEK (PL) <  > PV THEN
       POKE PL,PV:
      GOTO 20030
20060 POKE PL,127 - PV:
      GOTO 20030
20100 REM ---------------------
20110 REM ENCIENDE CURSOR XC,YC
20120 REM ---------------------
20125 IF XC < 1 OR XC > 40 THEN XC
       = 1
20127 IF YC < 1 OR YC > 24 THEN YC
       = 1
20130 POKE PL,PV
20140 LOC = 23423 * (YC < 9) + 23463
       * (YC > 8) * (YC < 17) + 23503
       * (YC > 16) + (YC - 8 * (YC
       > 8) - 8 * (YC > 16)) * 128
       + XC
20150 PV =  PEEK (LOC):
      PL = LOC:
      POKE LOC,127 - PV
20160 RETURN
20200 REM ------------------
20210 REM CARACTER EN HI-RES
20220 REM ------------------
20230 IO = (24575 + (CO - 32) * 8)
       + GT * 512
20240 LOC = 16255 * (Y < 9) + 16295
       * (Y > 8) * (Y < 17) + 16335
       * (Y > 16) + (Y - 8 * (Y > 8)
       - 8 * (Y > 16)) * 128 + X
20250 FOR I = 1 TO 8
20260    POKE LOC, PEEK (IO + I)
20270    LOC = LOC + 1024
20280 NEXT I
20290 RETURN
20300 REM ----------------
20310 REM CADENA EN HI-RES
20320 REM ----------------
20325 IF A$ = "" THEN  RETURN
20330 LL =  LEN (A$):
      FOR IL = 1 TO LL
20340    CO =  ASC ( MID$ (A$,IL,1)):
         GOSUB 20230
20350    X = X + 1:
      NEXT IL
20360 RETURN
20800 REM -----------------------
20810 REM CUADRADO A XC,YC DE N*M
20820 REM -----------------------
20830 X = (XC - 1) * 7:
      Y = (YC - 1) * 8
20840 L1 = 7 * N:
      L2 = 8 * M
20850 HPLOT X - 1,Y - 1 TO X + L1,Y
       - 1 TO X + L1,Y + L2 TO X
       - 1,Y + L2 TO X - 1,Y - 1
20860 RETURN
35000 REM -------------
35010 REM INIC.PANTALLA
35020 REM -------------
35030 HGR2 :
      HCOLOR= 3:
      GT = 0
35040 N = 5:
      M = 5:
      YC = 3
35050 XC = 5:
      GOSUB 20800
35060 XC = 14:
      GOSUB 20800
35070 YC = 11:
      GOSUB 20800
35080 XC = 23:
      GOSUB 20800
35090 XC = 5:
      GOSUB 20800
35[illegible] A$ = "SIMBOLO=":
      X = 32:
      Y = 1
35[illegible] GOSUB 20300:
      A$ = "SET=0"
35[illegible] X = 36:
      Y = 3:
      GOSUB 20300
35130 X = 1:
      Y = 1:
      A$ = "TEST N." +  STR$ (TE)
35140 GOSUB 20300:
      X = 20:
      Y = 5
35150 A$ = "<< PREGUNTA":
      GOSUB 20300
35160 X = 29:
      Y = 13:
      A$ = "<< RESPUESTA"
35170 GOSUB 20300:
      Y = 9:
      X = 7:
      A$ = "1"
35180 GOSUB 20300:
      X = 16:
      A$ = "2"
35190 GOSUB 20300:
      Y = 17:
      X = 7:
      A$ = "3"
35200 GOSUB 20300:
      X = 16:
      A$ = "4"
35210 GOSUB 20300:
      X = 25:
      A$ = "5"
35220 GOSUB 20300
35250 FOR I = 1 TO 5:
         FOR J = 1 TO 5:
            FOR K = 1 TO 5
35260          CC%(I,J,K) = 32:
               GG%(I,J,K) = 0
35270       NEXT K:
         NEXT J:
      NEXT I:
      RETURN
35500 REM ----------------
35510 REM CONSTRUC.SPRITES
35520 REM ----------------
35525 DATA 5,3,14,3,5,11,14,11,23,11
35530 SP = 1:
      RESTORE :
      CC = 32:
      GT = 0
35540 FOR I = 1 TO 5:
         READ XI%(I),YI%(I):
      NEXT I
35550 XC = XI%(SP):
      YC = YI%(SP)
35560 GOSUB 20100
35570 GOSUB 20000:
      A =  FRE (0)
35580 IF A0 <  > 1 THEN 35610
35590 IF XC = XI%(SP) THEN 35570
35600 XC = XC - 1:
      GOTO 35560
35610 IF A0 <  > 19 THEN 35640
35620 IF XC = XI%(SP) + 4 THEN 35570
35630 XC = XC + 1:
      GOTO 35560
35640 IF A0 <  > 23 THEN 35670
35650 IF YC = YI%(SP) THEN 35570
35660 YC = YC - 1:
      GOTO 35560
35670 IF A0 <  > 26 THEN 35700
35680 IF YC = YI%(SP) + 4 THEN 35570
35690 YC = YC + 1:
      GOTO 35560
35700 IF A0 <  > 13 THEN 35760
35710 CO = CC:
      X = XC:
      Y = YC:
      GOSUB 20200
35720 PV =  PEEK (LOC - 1024)
35730 CC%(XC - XI%(SP) + 1,YC - YI%(SP)
       + 1,SP) = CC
35740 GG%(XC - XI%(SP) + 1,YC - YI%(SP)
       + 1,SP) = GT
35750 GOTO 35570
```

```
35760 IF A0 <  > 8 THEN 35800
35770 IF CC = 32 THEN CC = 97
35780 CC = CC - 1:
      C0 = CC:
      X = 40:
      Y = 1
35790 GOSUB 20200:
      GOTO 35570
35800 IF A0 <  > 21 THEN 35840
35810 IF CC = 96 THEN CC = 31
35820 CC = CC + 1:
      C0 = CC:
      X = 40:
      Y = 1
35830 GOSUB 20200:
      GOTO 35570
35840 IF A0 <  > 32 THEN 35880
35850 IF GT = 3 THEN GT =  - 1
35860 GT = GT + 1:
      GG = GT:
      C0 = CC:
      X = 40:
      Y = 1
35870 GOSUB 20200:
      GT = 0:
      X = 40:
      Y = 3:
      A$ =  STR$ (GG):
      GOSUB 20300:
      GT = GG:
      GOTO 35570
35880 IF A0 > 48 AND A0 < 54 THEN SP
       = A0 - 48:
      GOTO 35550
35890 IF A0 <  > 24 THEN 35920
35900 C0 = 32:
      X = XC:
      Y = YC:
      GOSUB 20200:
      PV =  PEEK (LOC - 1024)
35910 CC%(XC - XI%(SP) + 1,YC - YI%(SP)
       + 1,SP) = C0:
      GOTO 35570
35920 IF A0 <  > 5 THEN 35570
35925 GT = 0
35930 A$ = "SEGURO ? (S/N)":
      X = 13:
      Y = 19
35940 GOSUB 20300:
      GET C$:
      IF C$ <  > "S" AND C$ <  > "N"
       THEN 35930
35950 A$ = "                   ":
      X = 13:
      GOSUB 20300
35960 IF C$ <  > "S" THEN 35570
35970 FOR SP = 1 TO 5:
         AA$ = ""
35980    FOR J = 1 TO 5
35990       FOR I = 1 TO 5
36000          AA$ = AA$ +  STR$ (CC%(I
               ,J,SP))
36010          AA$ = AA$ +  STR$ (GG%(I
               ,J,SP))
36020       NEXT I:
         NEXT J
36030    NR = IR + SP:
         GOSUB 6400:
      NEXT SP
36040 RETURN
```

Símbolos-Uso

```
10    REM    -----------------
20    REM    TEST-SIMBOLOS USO
30    REM       VERSION  DOS
40    REM    -----------------
50    D$ =  CHR$ (4):
      BP$ =  CHR$ (7)
55    ONERR  GOTO 40000
60    B$ = "          ":
      BA$ = B$ + B$ + B$ + B$:
      B$ = ""
65    DATA   6,3,15,3,6,15,15,15,24,15
70    PRINT D$:
      PRINT D$;"BLOAD CAR,D1":
      PRINT D$
80    HGR2 :
      HCOLOR= 3
85    RESTORE :
      FOR I = 1 TO 5:
         READ XS(I),YS(I):
      NEXT I
90    X = 7:
      Y = 13:
      A$ = "CUANTAS UNIDADES HAY ? (1/2
      )":
      GOSUB 20300
100   GET C$:
      IF C$ <  > "1" AND C$ <  > "2"
       THEN 90
110   DR$ = C$:
      A$ = "INTRODUCIR DISCO DATOS EN L
      A UNIDAD " + C$
120   X = 3:
      GT = 1:
      GOSUB 20300
130   A$ = "PULSAR UNA TECLA":
      GT = 0
140   X = 12:
      Y = 23:
      GOSUB 20300
150   GET C$
160   NF$ = "SIMBOLOS":
      LR = 75:
      GOSUB 6000
170   NR = 1:
      GOSUB 6200:
      MX =  VAL (AA$)
180   IR = 2:
      JK = 1
190   GOSUB 35000
200   GOSUB 35500
210   FOR SP = 1 TO 5
220      XS = XS(SP):
         YS = YS(SP):
         GOSUB 36000
230   NEXT SP:
      GT = 0
240   A$ = "CUAL ? (1,2,3)":
      X = 14:
      Y = 24
250   GOSUB 20300
260   GET C$:
      IF C$ < "1" OR C$ > "3" THEN 260
270   IF  VAL (C$) = KK - 2 THEN 300
275   IF A$ = "E" THEN  PRINT BP$:
      GOTO 260
280   PRINT BP$;:
      A$ = "ERROR, VUELVE A PROBAR":
      GT = 1
290   X = 14:
      Y = 24:
      GOSUB 20300:
      A$ = "E":
      GOTO 260
300   HGR2 :
      A$ = "MUY BIEN, TEST SUPERADO":
      GT = 1
310   X = 9:
      Y = 13:
      GOSUB 20300
320   A$ = "OTRO TEST ? (S/N)":
      GT = 0:
      Y = 23
330   X = 13:
      GOSUB 20300
340   GET C$:
      IF C$ <  > "S" AND C$ <  > "N"
       THEN 340
350   IF C$ = "N" THEN  TEXT :
      HOME :
      END
360   IF JK = MX THEN  HGR2 :
      A$ = "TESTS TERMINADOS":
      GT = 1:
      X = 14:
      Y = 13:
      GOSUB 20300:
      GET C$:
      TEXT :
      HOME :
      END
370   JK = JK + 1:
      IR = IR + 6:
      GOTO 190
6000  REM  -----------------
6010  REM  APERTURA FICHEROS
6020  REM  -----------------
6030  PRINT D$
6040  PRINT D$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D$
6060  RETURN
6200  REM  ----------------
6210  REM  LECTURA REGISTRO
6220  REM  ----------------
6230  PRINT D$
6240  PRINT D$;"READ";NF$;",R";NR
6250  INPUT AA$
6260  PRINT D$
6270  RETURN
6600  REM  --------------
6610  REM  CIERRA FICHERO
6620  REM  --------------
6630  PRINT D$
6640  PRINT D$;"CLOSE";NF$
6650  PRINT D$
6660  RETURN
20200 REM ------------------
20210 REM CARACTER EN HI-RES
20220 REM ------------------
20230 I0 = (24575 + (C0 - 32) * 8)
       + GT * 512
20240 LOC = 16255 * (Y < 9) + 16295
       * (Y > 8) * (Y < 17) + 16335
       * (Y > 16) + (Y - 8 * (Y > 8)
       - 8 * (Y > 16)) * 128 + X
20250 FOR I = 1 TO 8
20260    POKE LOC, PEEK (I0 + I)
20270    LOC = LOC + 1024
20280 NEXT I
20290 RETURN
20300 REM ----------------
20310 REM CADENA EN HI-RES
20320 REM ----------------
20325 IF A$ = "" THEN  RETURN
20330 LL =  LEN (A$):
      FOR IL = 1 TO LL
20340    C0 =  ASC ( MID$ (A$,IL,1)):
         GOSUB 20230
20350    X = X + 1:
      NEXT IL
20360 RETURN
20800 REM -----------------------
20810 REM CUADRADO A XC,YC DE N*M
20820 REM -----------------------
20830 X = (XC - 1) * 7:
      Y = (YC - 1) * 8
20840 L1 = 7 * N:
      L2 = 8 * M
20850 HPLOT X - 1,Y - 1 TO X + L1,Y
       - 1 TO X + L1,Y + L2 TO X
       - 1,Y + L2 TO X - 1,Y - 1
20860 RETURN
35000 REM -------------
```

```
35010 REM INIC.PANTALLA
35020 REM -------------
35030 HGR2 :
      N = 5:
      M = 5:
      YC = 3
35040 XC = 6:
      GOSUB 20800
35050 XC = 15:
      GOSUB 20800
35060 XC = 24:
      GOSUB 20800
35070 YC = 15:
      XC = 6:
      GOSUB 20800
35080 XC = 15:
      GOSUB 20800
35090 XC = 24:
      GOSUB 20800
35100 A$ = "COMPLETAR CON LA FIGURA COR
      RECTA."
35110 X = 3:
      Y = 12:
      GOSUB 20300
35120 A$ = "1":
      Y = 21:
      X = 8:
      GOSUB 20300
35130 A$ = "2":
      X = 17:
      GOSUB 20300
35140 A$ = "3":
      X = 26:
      GOSUB 20300
35170 Y = 5:
      X = 26:
      A$ = "?":
      GOSUB 20300
35180 RETURN
35500 REM -------------------
35510 REM LECTURA SPRITE DISCO
35520 REM -------------------
35530 FOR IS = IR TO IR + 4
35540    NR = IS:
         GOSUB 6200
35550    X = 1:
         Y = 1:
         FOR I = 1 TO 73 STEP 3
35560        CO =  VAL ( MID$ (AA$,I,2))
35570        GT =  VAL ( MID$ (AA$,I
          + 2,1))
35580        CC%(X,Y,IS - IR + 1)
          = CO
35590        GG%(X,Y,IS - IR + 1)
          = GT
35600        X = X + 1:
             IF X > 5 THEN X = 1:
             Y = Y + 1
35610    NEXT I:
      NEXT IS
35620 NR = NR + 1:
      GOSUB 6200:
      KK =  VAL (AA$)
35630 RETURN
36000 REM -------------
36010 REM DIBUJA SPRITE
36020 REM -------------
36030 FOR Y = YS TO YS + 4
36040    FOR X = XS TO XS + 4
36050       CO = CC%(X - XS + 1,Y
          - YS + 1,SP)
36060       GT = GG%(X - XS + 1,Y
          - YS + 1,SP)
36070       GOSUB 20200
36080    NEXT X:
      NEXT Y
36090 RETURN
40000 TEXT :
      HOME
40010 IF  PEEK (222) = 5 THEN  PRINT "L
      OS DATOS NO EXISTEN":
      END
```

RELACION DE LAS PRINCIPALES VARIABLES - APPLE

Preguntas

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud del campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en pantalla del campo de introducción
DX(*) **DY(*)**	= coordenadas en pantalla de descripción
TF(*)	= matriz numérica con los códigos ASCII de teclas de comando
D$(*)	= descripciones en máscara video
VO$(*)	= voces del menú principal
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica campos de intr. alfanuméricos
NS$	= carácter "#"; identifica campos de intr. numéricos
HO$	= borrado de pantalla
NUC	= número de campos de intr. en la máscara video activada
DS	= número de descripciones en máscara video
AA$	= buffer del disco
BA$	= 80 blanks
BP$	= CHR$(7) [emite un beep]
CC	= contador campos del data entry
DR$	= unidad seleccionada
FS	= código tecla función pulsada

Atributos

AA$	= buffer del disco
B$(*)	= buffer del data entry
BA$	= 80 espacios

BL$	= blank
BP$	= CHR$(7) [beep]
CC	= contador campos del data entry
CX(*)	= coorden. X campos
CY(*)	= coorden. Y campos
D$(*)	= descripciones campos
DR$	= unidad seleccionada
DX(*)	= coordenadas X descripciones
DY(*)	= coordenadas Y descripciones
FS	= código tecla función pulsada
LC(*)	= longitud campos
LR	= longitud registro
MP$	= nombre programa
NC	= número campos
ND(*)	= número decimales
NF$	= nombre fichero
TF(*)	= códigos teclas función
TP(*)	= tipos de campos
MX	= máx. núm. de registros del fichero
NR	= número del registro seleccionado

Símbolos

BA$	= 80 blanks
AA$	= buffer del disco
GT	= número juego caracteres seleccionado
NF$	= nombre fichero
LR	= longitud registro
NR	= número del registro seleccionado
TE	= número del test seleccionado
CC(*)	= códigos carácter usados en dibujo HI-RES
GG(*)	= códigos juegos de caracteres del dibujo
MX	= número de registros del fichero
XS(*)	= coordenadas X del sprite (dibujo)
YS(*)	= coordenadas Y del sprite (dibujo)
C0	= código carácter HI-RES a imprimir

Test: lógica y leyes asociativas/Versión Philips VG 8010

MSX

Puntuación-Generación

```
10  REM  ------------------------
20  REM  T E S T  con  PUNTUACION
25  REM        generacion
27  REM        VERSION  MSX
30  REM  ------------------------
35  KEY OFF:KEY 1,CHR$(15):KEY 2,CHR$(16):KEY 3,CHR$(17):KEY 4,CHR$(19):KEY5,CHR$(20):WIDTH 40:C
LS:CLEAR6000
40  DIM TF(15),TP(20),LC(20),ND(20),CX(20),CY(20),DX(20),DY(20)
50  DIM B$(20),D$(20),TS$(20),RI$(10)
60  REM  CONSTANTES
70  A1=1:A2=31:BP$="08L30DFB":NS$="=":AS$="-":PD$=".":BL$=" "
75  F$=CHR$(1)+CHR$(87)
77  Q$="":FOR I=0 TO 39:Q$=Q$+F$: NEXT I
80  SZ=2:MI=1
100 US$=CHR$(13):ZE$=CHR$(48):GIU$=CHR$(32)
110 HE$="CTRL Y (AYUDA)"
115 H1$="E.G.S.          Ediciones Forum, S.A."
120 REM  CARGA VOCES
130 RESTORE
140 READ PROG$
190 FOR I=1 TO 15
200 READ TF(I)
210 NEXT I
219 REM -----------
220 REM   M A I N
230 REM -----------
250 OPZ=1:GOSUB 7000
260 GOSUB 2000:X=10:Y=8:GOSUB11000:PRINT"INTRODUCIR LOS DATOS:"
270 GOSUB 3000
280 IF FS<>5 THEN 260
290 MX=VAL(B$(2)):IF MX>20 OR MX<1 THEN 260
300 CLS:X=6:Y=8:GOSUB11000:PRINT"Posicionar la cinta datos":Y=10:GOSUB11000:PRINT" y apretar REC
 y PLAY."
310 X=2:Y=21:GOSUB 11000:PRINT"NOTA: Dejar el grabador en esta":Y=22:GOSUB 11000:PRINT"posicion
hasta el fin del programa.":X=6:Y=14:GOSUB 11000:PRINT"HECHO ? (pulsa una tecla)"
320 IF INKEY$="" THEN 320
330 GOSUB 1000:OPEN"CAS:az"FOROUTPUTAS#1
332 A$="":FORI=1TO6:PRINT#1,A$:NEXT:CLOSE#1:REM espacio cinta
335 OPEN"CAS:INIZ"FOROUTPUTAS#1
340 A$=B$(1)+B$(2):PRINT#1,A$:CLOSE#1
350 FOR JK=1 TO MX: REM bucle principal
360 OPZ=2:GOSUB7000
380 FOR JJ=0 TO 1:GOSUB 2000
390 IF JJ=0 THEN T$=" (Primera" ELSE T$="(Segunda"
400 X=7:Y=6:GOSUB 11000:PRINT"INTRODUCIR LA PREGUNTA N.";JK
410 X=12:Y=8:GOSUB11000:PRINTT$;" parte)"
420 GOSUB 3000
430 IF FS=6 THEN 380 ELSE IF FS=7 THEN 230
440 FOR I=1 TO 10
450 IF LEFT$(B$(I),20)=SPACE$(20) THEN B$(I)=""
460 TS$(I+JJ*10)=B$(I):NEXT I
500 NEXT JJ
510 OPZ=3:GOSUB7000
520 GOSUB 2000
530 X=7:Y=6:GOSUB11000:PRINT"RESPUESTAS A LA PREGUNTA N.";JK
535 X=2:Y=8:GOSUB11000:PRINT"Respuesta                      Puntos";
540 GOSUB 3000
550 IF FS=6 THEN 520 ELSE IF FS=7 THEN 230
560 A=0:FOR I=1 TO 19 STEP2:A=A+1
565 IF LEFT$(B$(I),20)=SPACE$(20) THEN B$(I)="":B$(I+1)=""
570 RI$(A)=B$(I)+B$(I+1):NEXTI
600 REM grabacion
605 GOSUB 1000
610 OPEN"CAS:PRE"FOROUTPUTAS#1
620 FOR I=1 TO 20:PRINT#1,TS$(I)
630 NEXT I:CLOSE#1
640 OPEN"CAS:RES"FOROUTPUTAS#1
650 FORI=1TO10:PRINT#1,RI$(I)
660 NEXTI:CLOSE#1
670 NEXTJK
700 OPZ=4:GOSUB 7000
710 GOSUB 2000:X=12
720 FOR Y=10 TO 19:GOSUB 11000:Y$=STR$(Y-9)
730 PRINT RIGHT$(Y$,2)+":":NEXT Y
740 X=4:Y=6:GOSUB 11000:PRINT"INTRODUCIR LAS GAMAS DE PUNTUACION":X=15:Y=8:GOSUB 11000:PRINT"MIN
    MAX"
750 GOSUB 3000:IF FS=6 THEN 710
760 IFFS=7THEN230
770 FOR I=1 TO 20:IF VAL(B$(I))<10 THEN B$(I)="0"+RIGHT$(B$(I),1)
780 TS$(I)=B$(I):NEXT I
800 OPZ=5:GOSUB 7000:DY(3)=DY(3)-2
810 WW=0:FOR JK=1 TO 19 STEP2:WW=WW+1
815 IF TS$(JK)="0=" OR TS$(JK)="0 " THEN RI$(WW)="":GOTO 860
820 B$(1)=TS$(JK):B$(2)=TS$(JK+1)
830 GOSUB 2000:GOSUB 3000
840 IF FS=6 THEN 830 ELSE IF FS=7 THEN 230
850 RI$(WW)=B$(3)+B$(4)+B$(5)
860 NEXT JK
900 REM GRABACION
905 GOSUB 1000
910 OPEN"CAS:FA"FOROUTPUTAS#1
920 FOR I=1 TO 20:PRINT#1,TS$(I):NEXT I
930 CLOSE#1
940 OPEN"CAS:DI"FOROUTPUTAS#1
950 FOR I=1 TO 10:PRINT#1,RI$(I):NEXT I
960 CLOSE#1
970 CLS:X=12:Y=11:GOSUB 11000:PRINT"FIN PROGRAMA":Y=20:GOSUB 11000
980 END
1000 REM -------
1003 REM ESPERAR
1005 REM -------
1010 CLS:X=15:Y=11:GOSUB 11000
1020 PRINT"E S P E R A R"
1030 RETURN
2000 REM ----------------
2010 REM PRESENTA MASCARA
2020 REM ----------------
2030 CLS:COLOR4,7
2130 L=LEN(PROG$)
2140 FM=(38-L)/2:FM$=SPACE$(FM)
2150 X=1:Y=0:GOSUB11000:PRINTH1$;
2215 Y=3:X=0:GOSUB 11000:PRINT FM$;"|"; PROG$;"|"; FM$:G8$=CHR$(1)+CHR$(&H57):X=FM:Y=2
2216 GOSUB 11000:G8$=CHR$(1)+CHR$(&H57):FOR J=1 TO(L-1):G8$=G8$+CHR$(1)+CHR$(&H57):NEXT:PRINTCHR
$(1);CHR$(&H58);G8$;CHR$(1);CHR$(&H59):Y=4:GOSUB 11000:PRINTCHR$(1);CHR$(&H5A);G8$;CHR$(1);CHR$(
&H5B)
2218 Y=21:X=0:GOSUB 11000:PRINT Q$
2219 Y=22:X=25:GOSUB 11000
2220 PRINT HE$
2240 FOR J=1 TO NUC
2245 A$= ""
2270 X=DX(J):Y=DY(J):GOSUB 11000:PRINT D$(J)
2275 IF TP(J)<>5 AND TP(J)<>4 THEN 2290
2280 X=CX(J):Y=CY(J):GOSUB 11000
2285 PRINT B$(J):GOTO 2420
2290 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J) = 2 THEN C$=NS$:GOTO 2310
2300 C$=AS$
2310 A$=STRING$(N1-N2-1,C$)
2320 IF N2=0 THEN A$= A$+C$:GOTO 2350
2330 A$=A$+PD$
2340 FOR I=1 TO N2:A$=A$+C$:NEXT I
2350 X=CX(J):Y=CY(J):GOSUB 11000
2360 PRINT A$
2380 B$(J)= SPACE$ (N1-N2-1)
2410 B$(J)= B$(J)+RIGHT$ (A$,N2+1)
2420 NEXT J
2430 RETURN
3000 REM ----------
3010 REM DATA ENTRY
3020 REM ----------
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
```

```
3060 GOSUB 3240
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 PLAY BP$ :FS=0:J=NUC:GOTO 3040
3150 IF J< NUC THEN J=J+1:GOTO 3040
3160 PLAY  BP$:FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 PLAY  BP$:FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB 11000:GOTO 3060
3200 PLAY  BP$:FS=0:GOTO 3040
3210 IF FS>8 THEN FS=0:GOTO 3060
3220 IF FS=8 THEN FS=0:GOTO 3100
3230 RETURN
3240 REM -------------------
3250 REM LECTURA DE UN CAMPO
3260 REM -------------------
3270 FS=0
3280 IF TP(NC)=4 THEN RETURN
3300 IF MID$(B$(NC),CC,1)=PD$ANDTP(NC)<>3 THEN X=X+S:CC=CC+S
3310 GOSUB 11000
3320 A$=INKEY$:IF A$= ""ORA$="," THEN  PRINT"█";:GOSUB11000:PRINTMID$(B$(NC),CC,1);:GOSUB11000:G
OTO3320
3330 CA=ASC(A$): IF CA > A2 THEN 3380
3340 FOR I=1 TO 15
3350 IF TF(I)= CA  THEN FS = I: GOSUB 3630: RETURN
3360 NEXT I
3370 GOTO 3320
3380 IF TP(NC)<>1 THEN 3440
3390 IF CA=45 OR CA=46 OR (CA>=48 AND CA<=57) THEN 3410
3400 PLAY BP$:GOTO 3320
3410 IF CA <> 46 THEN 3440
3420 GOSUB 4270
3430 IF SWP=1 THEN 3400
3440 S=1: GOSUB 3520
3450 IF CC=LC(NC) THEN RETURN
3460 CC=CC+1:X=X+1:GOTO 3300
3520 REM -----------------------
3530 REM TRANSFERENCIA AL BUFFER
3540 REM -----------------------
3550 IF CC=1 THEN SN$= "":GOTO 3570
3560 SN$=LEFT$(B$(NC),CC-1)
3570 IF CC=LC(NC) THEN DS$="":GOTO 3590
3580 N=LC(NC)-CC:DS$=RIGHT$ (B$(NC),N)
3590 B$(NC)=SN$+A$+DS$
3620 PRINT A$:RETURN
3630 REM --------
3640 REM COMANDOS
3650 REM --------
3660 IF FS>5 THEN 3690
3680 RETURN
3690 IF FS=9 THEN GOSUB 13000: RETURN
3700 IF FS=8 THEN GOSUB 4080:RETURN
3710 IF FS<8 THEN RETURN
3720 ON FS-9 GOSUB 3740,3910
3730 RETURN
3740 REM ---------
3750 REM INSERCION
3760 REM ---------
3770 IF CC=LC(NC) THEN PLAY BP$:RETURN
3780 IF CC=1 THEN SN$= "":GOTO 3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DS$=RIGHT$ (B$(NC),N)
3810 IF TP(NC)=1 OR TP(NC)=2 THEN C$=BL$:GOTO 3830
3820 C$=BL$
3830 A$=SN$ +C$+DS$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 PRINT B$(NC):S=-1
3890 IF RIGHT$(B$(NC),1)= PD$ THEN 3780
3900 RETURN
3910 REM -------
3920 REM BORRADO
3930 REM -------
3940 IF TP(NC)<>1  THEN 3960
3950 IF MID$ (B$(NC),2,1) = PD$ AND CC=1 THEN RETURN
3960 IF CC<1 THEN PRINT BP$: RETURN
3970 IF CC=LC(NC) THEN DS$= "": GOTO 4000
3980 N=LC(NC) -CC:DS$ =RIGHT$ (B$(NC),N)
3990 IF CC=1 THEN SN$= "": GOTO 4010
4000 SN$=LEFT$  (B$(NC),CC-1)
4010 IF TP(NC) = 1 OR TP(NC)=2 THEN C$=NS$:GOTO 4030
4020 C$=AS$
4030 B$(NC)= SN$ +DS$ +C$
4040 XX=X:YY=Y
4050 X=CX(NC):Y=CY(NC):GOSUB 11000
4060 X=XX:Y=YY
4070 PRINT B$(NC);:S=1:RETURN
4080 REM ---------
4090 REM ALINEADOS
4100 REM ---------
4110 IF LC(NC) =1 THEN RETURN
4120 N1=LC(NC) -CC+1
4130 IF CC=1 THEN B$(NC) = "":GOTO 4150
4140 B$(NC)= LEFT$ (B$(NC),CC-1)
4150 IF TP(NC) =1 OR TP(NC)=2 THEN 4200
4160 FOR I=1 TO N1
4170 B$(NC)=B$(NC)+BL$
4180 NEXT I
4190 GOTO 4240
4200 FOR I=1 TO N1
4210 B$(NC)=BL$ +B$(NC)
4220 NEXT I
4230 IF RIGHT$ (B$(NC),1) =PD$ THEN B$(NC)= BL$ +LEFT$ (B$(NC),LC(NC)-1)
4240 X=CX(NC):Y=CY(NC)
4250 GOSUB 11000
4260 PRINT B$(NC):RETURN
4270 REM ----------
4280 REM CTRL.PUNTO
4290 REM ----------
4300 SWP = 0
4310 IF CC=1 OR CC=LC(NC) THEN SWP =1:RETURN
4320 FOR I=1 TO LC(NC)
4330 IF MID$(B$(NC),I,1)= PD$ THEN I = LC(NC): SWP=1: RETURN
4340 NEXT I: RETURN
7000 REM --------------------
7010 REM LECTURA DE LOS DATOS
7020 REM --------------------
7030 RESTORE 10080
7040 READ NV
7050 FOR I = 1 TO NV:READ NUC(I),QR(I)
7060 NEXT I
7065 NUC=NUC(OPZ)
7070 FOR K=0 TO QR(OPZ)-1
7071 ON OPZ GOSUB 7200,7210,7220,7230,7240
7072 FOR I=1 TO NUC
7074 READ D$(I+K)
7076 NEXT I
7080 FOR I= 1+K*NUC TO NUC+K*NUC
7090 READ TP(I),LC(I),ND(I),DX(I),CX(I),CY(I)
7100 CY(I)=CY(I)+K:DY(I)=CY(I)
7110 NEXT I,K
7115 NUC=NUC*QR(OPZ)
7120 RETURN
7200 RESTORE 10100:RETURN
7210 RESTORE 10160:RETURN
7220 RESTORE 10210:RETURN
7230 RESTORE 10270:RETURN
7240 RESTORE 10300:RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA " T E S T (generacion)"
10070 DATA 30,31,28,29,20,16,15,13,25,19,17,0,0,0,0
10080 DATA 5
10090 DATA 2,1,1,10,2,10,2,10,5,1
10100 DATA "Sigla del test","Numero de preguntas"
10110 DATA 3,2,0,8,28,12
10120 DATA 1,2,0,8,28,15
10160 DATA " ",3,30,0,4,4,10
10210 DATA " "," "
10220 DATA 3,30,0,2,2,10
```

```
10230 DATA 1,2,0,35,35,10
10270 DATA " "," "
10280 DATA 1,2,0,16,16,10
10290 DATA 1,2,0,26,26,10
10300 DATA "Puntuacion","-","RESULTADO:"," "," "
10310 DATA 4,2,0,6,16,8
10320 DATA 4,2,0,19,21,8
10330 DATA 3,27,0,11,6,16
10340 DATA 3,27,0,0,6,17
10350 DATA 3,27,0,0,6,18
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 LOCATE X,Y
11040 RETURN
13000 REM -----
13010 REM AYUDA
13020 REM -----
13050 LOCATE 0,22
13055 PRINT "AYUDA (barra espaciadora para continuar)";
13060 GOSUB 13600
13110 PRINT "FLECHAS: mueven en las 4 direcciones    ";
13120 GOSUB 13600
13130 PRINT "RETURN: alinea y convalida introduccion ";
13140 GOSUB 13600
13150 PRINT "F1: anula y recomienza el programa      ";
13160 GOSUB 13600
13170 PRINT "F2: anula toda la introduccion          ";
13180 GOSUB 13600
13190 PRINT "F3: borra un caracter                   ";
13200 GOSUB 13600
13210 PRINT "F4: permite la insercion de espacios    ";
13212 GOSUB 13600
13215 PRINT "F5: graba toda la introduccion          ";
13220 GOSUB 13600
13230 PRINT "                         CTRL Y (AYUDA) ";
13300 RETURN
13600 A$=INKEY$:IF A$= "" THEN 13600
13605 IF A$<> CHR$(32) THEN 13600
13610 LOCATE 0,22
13620 RETURN
```

Puntuación-Uso

```
10 REM -------------------
20 REM TEST con PUNTUACION
25 REM      U S O
26 REM    Version MSX
30 REM -------------------
31 MAXFILES=2
32 SCREEN2:COLOR4,15,7:CLS:CLEAR 3000
33 OPEN"GRP:"FOROUTPUTAS#1
35 KEY OFF
50 DIM TS$(20),RI$(10)
70 BP$="O5L32DFB"
75 F$=CHR$(1)+CHR$(87)
77 Q$="":FOR I=1 TO 32:Q$=Q$+F$: NEXT I
110 HE$="CTRL Y (AYUDA)"
115 H1$="E.G.S.   Ediciones Forum, S.A."+Q$
120 REM CARGA VOCES
130 RESTORE
140 READ PROG$
210 REM
218 REM ----
220 REM MAIN
230 REM ----
232 REM
233 GOSUB2000
235 X=2:Y=8:GOSUB 11000:PRINT#1,"Posicionar la cinta datos":K=6:X=7:Y=11:GOSUB 11100:PRINT#1,"y
apretar PLAY."
240 K=4:X=3:Y=14:GOSUB 11100:PRINT#1,"Hecho ? (pulsar una tecla)"
245 K=2:X=1:Y=20:GOSUB 11100:PRINT#1,"NOTA:dejar PLAY apretado          durante todo el progr
ama."
250 IF INKEY$="" THEN 250
260 GOSUB 1000
270 OPEN"CAS:INIZ"FORINPUTAS#2
280 INPUT#2,A$:MX=VAL(MID$(A$,3,2))
290 CLOSE#2
300 CLS:GOSUB 2000:X=4:Y=9:K=1:GOSUB11100:PRINT#1,"Esta cinta contiene":X=10:Y=11:GOSUB11000:PRI
NT#1,"el test ";LEFT$(A$,2)
310 X=8:Y=20:K=3:GOSUB 11100:PRINT#1,"PULSA UNA TECLA":PLAYBP$+BP$
320 IFINKEY$=""THEN320
350 REM PRINCIPIO BUCLE
360 FOR JK=1 TO MX:GOSUB 1000
370 OPEN"CAS:PRE"FORINPUTAS#2
380 FOR I=1 TO 20:INPUT#2,TS$(I):NEXT
390 CLOSE#2
400 OPEN"CAS:RES"FORINPUTAS#2
410 FORI=1TO10:INPUT#2,RI$(I):NEXT
420 CLOSE#2
440 CLS:K=6:X=8:Y=.5:GOSUB 11100:PRINT#1,"HE AHI EL TEST:"
450 FOR Y=2 TO 21:X=1:K=4:GOSUB 11100:PRINT#1,TS$(Y-1):NEXT
460 X=8:Y=23:K=1:GOSUB11100:PRINT#1,"pulsar una tecla":PLAYBP$
470 IF INKEY$="" THEN 470
480 CLS:GOSUB2000:X=4:Y=7:K=6:GOSUB11100:PRINT#1,"ELEGIR LA SOLUCION:"
485 FOR Y=9 TO 18:IF MID$(RI$(Y-8),31,4)="" THEN 495
490 X=.3:K=4:GOSUB11100:PRINT#1,RIGHT$(STR$(Y-9),1);")";LEFT$(RI$(Y-8),30)
495 NEXT
500 X=10:Y=22:K=6:GOSUB 11100:PRINT#1,"Cual ? _":PLAYBP$
510 I$=INKEY$:IF I$="" OR I$<CHR$(48) OR I$>CHR$(57) THEN 510 ELSE P=1+VAL(I$)
515 IF MID$(RI$(P),31,4)="" THEN 510
520 PP=PP+VAL(MID$(RI$(P),31,4))
530 NEXT JK
550 GOSUB 1000
560 OPEN"CAS:FA"FORINPUTAS#2
570 FOR I=1 TO 20:INPUT#2,TS$(I):NEXT
580 CLOSE#2
590 OPEN"CAS:DI"FORINPUTAS#2
600 FOR I=1 TO 10:INPUT#2,RI$(I):NEXT
610 CLOSE#2
620 A=0:FOR I=1 TO 19 STEP2:A=A+1
630 MI(A)=VAL(TS$(I)):MX(A)=VAL(TS$(I+1)):NEXT
640 FOR IJ=1 TO 10
650 IF MX(IJ)=0 THEN 670
660 IF PP>=MI(IJ)ANDPP<=MX(IJ) THEN 700
670 NEXT IJ
680 CLS:X=1:Y=9:K=6:GOSUB11100:PRINT#1,"SE HA COMETIDO UN ERROR":Y=11:GOSUB 11000:PRINT#1,"EN LA
 FASE DE GENERACION;":Y=13:K=2:GOSUB11100:PRINT#1,"EL RESULTADO DEL TEST NO PUEDE PRESENTARSE."
690 IF INKEY$="" THEN 690 ELSE END
700 CLS:GOSUB 2000
710 X=8:Y=9:K=4:GOSUB 11100:PRINT#1,"RESULTADO:":K=6
720 A=1:FOR Y=12 TO 16 STEP2
730 X=2.5:GOSUB11100:PRINT#1,MID$(RI$(IJ),A,27):A=A+27
740 NEXT Y
750 X=8:Y=20:K=1:GOSUB11100:PRINT#1,"PULSA UNA TECLA":PLAYBP$+BP$,"O4L16BBBO5DDD"
760 GOTO690
1000 REM -------
1003 REM ESPERAR
1005 REM -------
1010 CLS:COLOR4,15:PRESET(55,90):PRINT#1,"E S P E R A R"
1020 RETURN
2000 REM ----------------
2010 REM PRESENTA MASCARA
2020 REM ----------------
2130 L=LEN(PROG$)
2140 FM=(32-L)/2:FM$=SPACE$(FM)
2150 X=1:Y=.5:K=2:GOSUB 11100:PRINT#1,H1$;:COLOR1
2215 GOSUB 11000:LINE((FM-1)*8,26)-((32-FM)*8,43),7,BF
2216 Y=4:X=0:GOSUB 11000:PRINT#1, FM$;PROG$;FM$:X=FM:Y=3
2217 RETURN
10030 DATA "T E S T  USO"
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 PRESET(X*8,Y*8)
11040 RETURN
11100 REM ----------------
11110 REM POS.CURSOR COLOR
11120 REM ----------------
11130 PRESET(X*8,Y*8)
11135 COLORK
11140 RETURN
```

Atributos-Generación

```
10 REM -----------------------
20 REM T E S T  con  ATRIBUTOS
25 REM        generacion
```

```
27 REM     Version  MSX
30 REM -----------------------
35 KEY OFF:KEY 1,CHR$(15):KEY 2,CHR$(16):KEY 3,CHR$(17):KEY 4,CHR$(19):KEY5,CHR$(20):WIDTH 40:C
LS:CLEAR10000
40 DIM TF(15),TP(20),LC(20),ND(20),CX(20),CY(20),DX(20),DY(20)
47 M3=100
50 DIM B$(20),D$(20),AT$(10),EL$(M3)
60 REM CONSTANTES
70 A1=1:A2=31:BP$="O6L30DFB":NS$="=":AS$="-":PD$=".":BL$=" "
75 F$=CHR$(1)+CHR$(87)
77 O$="":FOR I=0 TO 39:O$=O$+F$: NEXT I
80 SZ=2:MI=1
100 US$=CHR$(13):ZE$=CHR$(48):GIU$=CHR$(32)
110 HE$="CTRL Y (AYUDA)"
115 H1$="E.G.S            Ediciones Forum,S.A."
120 REM CARGA VOCES
130 RESTORE
140 READ PROG$
190 FOR I=1 TO 15
200 READ TF(I)
210 NEXT I
219 REM -------
220 REM M A I N
230 REM -------
240 REM
250 OPZ=1:GOSUB 7000
260 GOSUB 2000
270 GOSUB 3000
280 IF FS<>5 THEN 260
290 NT$=B$(1)
300 X=0:Y=11:GOSUB 305
302 Y=21:GOSUB305
303 Y=22:GOSUB305:GOTO310
305 GOSUB11000:PRINTSPACE$(40);:RETURN
310 X=8:Y=9:GOSUB11000:PRINT"1) Introduccion nuevo test.":Y=11:GOSUB11000:PRINT"2) Correccion te
st.":X=11:Y=16:GOSUB11000:PRINT"CUAL ? (1 o 2)"
320 I$=INKEY$:IF I$<>"2" AND I$<>"1" THEN 320
325 W7=VAL(I$)
327 IF W7=1 THEN 360
330 T$=" PLAY":GOSUB900
335 OPEN"CAS:ATT"FORINPUTAS#1
340 INPUT#1,NN$
342 IF NN$<>NT$ THEN CLS:X=7:Y=10:GOSUB11000:PRINT"ERROR: el nombre del test":Y=12:GOSUB 11000:P
RINT"contenido en la cinta es ";NN$:PLAYBP$:CLOSE#1:FOR I=1 TO 2000:NEXT:GOTO 230
345 FOR I=1 TO 10:INPUT#1,AT$(I):NEXTI:CLOSE#1
350 OPEN"CAS:EL"FORINPUTAS#1
353 INPUT#1,A$:H=VAL(A$)
355 FOR I=1 TO H:INPUT#1,EL$(I):NEXT:CLOSE#1
360 OPZ=2:GOSUB 7000
370 IF W7=2 THEN FOR I=1 TO 10:TP(I)=5:B$(I)=AT$(I):NEXT
380 GOSUB 2000
400 X=7:Y=6:GOSUB 11000:PRINT"INTRODUCIR LOS ATRIBUTOS"
410 FOR Y=10 TO 19:X=2:GOSUB 11000:PRINTRIGHT$(STR$(Y-9),2);")":NEXT Y
420 GOSUB 3000
430 IF FS=6 THEN 380 ELSE IF FS=7 THEN 230
440 FOR I=1 TO 10
450 IF LEFT$(B$(I),20)=SPACE$(20) THEN B$(I)=""
460 AT$(I)=B$(I):NEXTI
510 OPZ=3:GOSUB7000
520 D$(1)="Elemento:":TP(1)=3:LC(1)=20:DX(1)=6:CX(1)=16:DY(1)=9:CY(1)=9
523 FOR JJ=1 TO M3:REM bucle elementos
525 IF W7=2 THEN FOR I=1 TO 11:TP(I)=5:NEXT ELSE 530
527 B$(1)=LEFT$(EL$(JJ),20):FOR I=2 TO 11:B$(I)=MID$(EL$(JJ),I+19,1):NEXT
530 GOSUB 2000
531 IF JJ<10 THEN X=2 ELSE X=1
532 Y=9:GOSUB 11000:PRINT STR$(JJ);CHR$(&HF8)
535 FORY=11TO20:X=14:GOSUB11000:IFY=11THENPRINTCHR$(1);CHR$(&H58);CHR$(1);CHR$(&H57);" ";ELSEIFY
=20THENPRINTCHR$(1);CHR$(&H5A);CHR$(1);CHR$(&H57);" ";ELSEIFY=15THENX=2:GOSUB11000:PRINT"Indicad
ores:";CHR$(1);CHR$(&H53);"  ";ELSE PRINT"|  ";
537 PRINTRIGHT$(STR$(Y-10),2);")":NEXTY
540 X=1:Y=6:GOSUB11000:PRINT"INTRODUCCION ELEMENTOS E INDICADORES"
545 GOSUB 3000
550 IF FS=6 THEN 530 ELSE IF FS=7 THEN 230
555 FOR I=2 TO 11:IF B$(I)>"1" THEN B$(I)="0"
556 NEXT I
560 EL$(JJ)="":FOR I=1 TO 11
570 EL$(JJ)=EL$(JJ)+B$(I):NEXTI
580 CLS:X=8:Y=10:GOSUB 11000:PRINT"Otro elemento ? (S o N)"
585 I$=INKEY$:IFI$="s" OR I$="S"THENNEXTJJ:JJ=JJ-1:CLS:X=3:Y=11:GOSUB11000:PRINT"SE HA ALCANZADO
EL NUMERO MAXIMO      DE ELEMENTOS (";M3;")":PLAYBP$:FORI=1TO2000:NEXT:GOTO600
590 IF I$<>"n" AND I$<>"N" THEN 585
600 REM grabacion
605 T$="REC y PLAY":GOSUB900
610 OPEN"CAS:AZ"FOROUTPUTAS#1
620 A$="":FOR I=1 TO 6:PRINT#1,A$
630 NEXT I:CLOSE#1:REM espacio cinta
640 OPEN"CAS:ATT"FOROUTPUTAS#1
645 PRINT#1,NT$
650 FORI=1TO10:PRINT#1,AT$(I):NEXT:CLOSE#1
660 OPEN"CAS:EL"FOROUTPUTAS#1
665 PRINT#1,STR$(JJ)
670 FOR I=1 TO JJ:PRINT#1,EL$(I):NEXT:CLOSE#1
700 CLS:X=12:Y=11:GOSUB 11000:PRINT"FIN PROGRAMA":Y=20:GOSUB 11000
710 END
900 REM POSICIONA CINTA
910 CLS:X=7:Y=9:GOSUB 11000:PRINT"Posicionar la cinta datos":Y=11:GOSUB 11000:PRINT"y apretar ";
T$:Y=15:GOSUB11000:PRINT"HECHO ? (pulse una tecla)"
930 IF INKEY$=""THEN 930
1000 REM ESPERAR
1010 CLS:X=13:Y=11:GOSUB11000
1020 PRINT"E S P E R A R"
1030 RETURN
2000 REM ----------------
2010 REM PRESENTA MASCARA
2020 REM ----------------
2030 CLS:COLOR4,7
2130 L=LEN(PROG$)
2140 FM=(38-L)/2:FM$=SPACE$(FM)
2150 X=1:Y=0:GOSUB11000:PRINTH1$:
2215 Y=3:X=0:GOSUB 11000:PRINT FM$;"|"; PROG$;"|"; FM$:G8$="-":X=FM:Y=2
2216 GOSUB 11000:G8$="-":FOR J=1 TO (L-1):G8$=G8$+"-":NEXT:PRINTCHR$(1);CHR$(&H58);G8$;CHR$(1);C
HR$(&H59):Y=4:GOSUB 11000:PRINTCHR$(1);CHR$(&H5A);G8$;CHR$(1);CHR$(&H5B);
2218 Y=21:X=0:GOSUB 11000:PRINT O$
2219 Y=22:X=25:GOSUB 11000
2220 PRINT HE$
2240 FOR J=1 TO NUC
2245 A$= ""
2270 X=DX(J):Y=DY(J):GOSUB 11000:PRINT D$(J)
2275 IFTP(J)<>5ANDTP(J)<>4THEN2290
2280 X=CX(J):Y=CY(J):GOSUB11000
2285 PRINTB$(J):GOTO2420
2290 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J) = 2 THEN C$=NS$:GOTO 2310
2300 C$=AS$
2310 A$=STRING$(N1-N2-1,C$)
2320 IF N2=0 THEN A$= A$+C$:GOTO 2350
2330 A$=A$+PD$
2340 FOR I=1 TO N2:A$=A$+C$:NEXT I
2350 X=CX(J):Y=CY(J):GOSUB 11000
2360 PRINT A$
2380 B$(J)= SPACE$ (N1-N2-1)
2410 B$(J)= B$(J)+RIGHT$ (A$,N2+1)
2420 NEXT J
2430 RETURN
3000 REM ----------
3010 REM DATA ENTRY
3020 REM ----------
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 PLAY BP$ :FS=0:J=NUC:GOTO 3040
3150 IF J< NUC THEN J=J+1:GOTO 3040
3160 PLAY  BP$:FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 PLAY  BP$:FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB 11000:GOTO 3060
3200 PLAY  BP$:FS=0:GOTO 3040
3210 IF FS>8 THEN FS=0:GOTO 3060
3220 IF FS=8 THEN FS=0:GOTO 3100
3230 RETURN
3240 REM -------------------
```

```
3250 REM LECTURA DE UN CAMPO
3260 REM -------------------
3270 FS=0
3280 IF TP(NC)=4 THEN RETURN
3290 IF TP(NC)=2 THEN GOSUB 9000:RETURN
3300 IF MID$(B$(NC),CC,1)=PD$ANDTP(NC)<>3 THEN X=X+S:CC=CC+S
3310 GOSUB 11000
3320 A$=INKEY$:IF A$= ""ORA$="," THEN  PRINT"█";:GOSUB11000:PRINTMID$(B$(NC),CC,1);:GOSUB11000:G
OTO3320
3330 CA=ASC(A$): IF CA > A2 THEN 3380
3340 FOR I=1 TO 15
3350 IF TF(I)= CA  THEN FS = I: GOSUB 3630: RETURN
3360 NEXT I
3370 GOTO 3320
3380 IF TP(NC)<>1 THEN 3440
3390 IF CA=45 OR CA=46 OR (CA>=48 AND CA<=57) THEN 3410
3400 PLAY BP$:GOTO 3320
3410 IF CA <> 46 THEN 3440
3420 GOSUB 4270
3430 IF SWP=1 THEN 3400
3440 S=1: GOSUB 3520
3450 IF CC=LC(NC) THEN RETURN
3460 CC=CC+1:X=X+1:GOTO 3300
3520 REM -----------------------
3530 REM TRANSFERENCIA AL BUFFER
3540 REM -----------------------
3550 IF CC=1 THEN SN$= "":GOTO 3570
3560 SN$=LEFT$(B$(NC),CC-1)
3570 IF CC=LC(NC) THEN DS$= "":GOTO 3590
3580 N=LC(NC)-CC:DS$=RIGHT$ (B$(NC),N)
3590 B$(NC)=SN$+A$+DS$
3620 PRINT A$:RETURN
3630 REM --------
3640 REM COMANDOS
3650 REM --------
3660 IF FS>5 THEN 3690
3680 RETURN
3690 IF FS=9 THEN GOSUB 13000: RETURN
3700 IF FS=8 THEN GOSUB 4080:RETURN
3710 IF FS<8 THEN RETURN
3720 ON FS-9 GOSUB 3740,3910
3730 RETURN
3740 REM ---------
3750 REM INSERCION
3760 REM ---------
3770 IF CC=LC(NC) THEN PLAY BP$:RETURN
3780 IF CC=1 THEN SN$= "":GOTO 3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DS$=RIGHT$ (B$(NC),N)
3810 IF TP(NC)=1 OR TP(NC)=2 THEN C$=BL$:GOTO 3830
3820 C$=BL$
3830 A$=SN$ +C$+DS$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 PRINT B$(NC):S=-1
3890 IF RIGHT$(B$(NC),1)= PD$ THEN 3780
3900 RETURN
3910 REM -------
3920 REM BORRADO
3930 REM -------
3940 IF TP(NC)<>1  THEN 3960
3950 IF MID$ (B$(NC),2,1) = PD$ AND CC=1 THEN RETURN
3960 IF CC<1 THEN PRINT BP$: RETURN
3970 IF CC=LC(NC) THEN DS$= "": GOTO 4000
3980 N=LC(NC) -CC:DS$ =RIGHT$ (B$(NC),N)
3990 IF CC=1 THEN SN$= "": GOTO 4010
4000 SN$=LEFT$  (B$(NC),CC-1)
4010 IF TP(NC) = 1 OR TP(NC)=2 THEN C$=NS$:GOTO 4030
4020 C$=AS$
4030 B$(NC)= SN$ +DS$ +C$
4040 XX=X:YY=Y
4050 X=CX(NC):Y=CY(NC):GOSUB 11000
4060 X=XX:Y=YY
4070 PRINT B$(NC);:S=1:RETURN
4080 REM ---------
4090 REM ALINEADOS
4100 REM ---------
4110 IF LC(NC) =1 THEN RETURN
4120 N1=LC(NC) -CC+1
4130 IF CC=1 THEN B$(NC) = "": GOTO 4150
4140 B$(NC)= LEFT$ (B$(NC),CC-1)
4150 IF TP(NC) =1 OR TP(NC)=2 THEN 4200
4160 FOR I=1 TO N1
4170 B$(NC)=B$(NC)+BL$
4180 NEXT I
4190 GOTO 4240
4200 FORI=1 TO N1
4210 B$(NC)=BL$ +B$(NC)
4220 NEXT I
4230 IF RIGHT$ (B$(NC),1) =PD$ THEN B$(NC)= BL$ +LEFT$ (B$(NC),LC(NC)-1)
4240 X=CX(NC):Y=CY(NC)
4250 GOSUB 11000
4260 PRINT B$(NC):RETURN
4270 REM ----------
4280 REM CTRL.PUNTO
4290 REM ----------
4300 SWP = 0
4310 IF CC=1 OR CC=LC(NC) THEN SWP =1:RETURN
4320 FOR I=1 TO LC(NC)
4330 IF MID$(B$(NC),I,1)= PD$ THEN I = LC(NC): SWP=1: RETURN
4340 NEXT I: RETURN
7000 REM --------------------
7010 REM LECTURA DE LOS DATOS
7020 REM --------------------
7030 RESTORE 10080
7040 READ NV
7050 FOR I = 1 TO NV:READ NUC(I),OR(I)
7060 NEXT I
7065 NUC=NUC(OPZ)
7070 FORK=0TOOR(OPZ)-1
7071 ON OPZ GOSUB 7200,7210,7220
7072 FOR I=1 TO NUC
7074 READ D$(I+K)
7076 NEXT I
7080 FOR I= 1+K*NUC TO NUC+K*NUC
7090 READ TP(I),LC(I),ND(I),DX(I),CX(I),CY(I)
7100 CY(I)=CY(I)+K:DY(I)=CY(I)
7110 NEXT I,K
7115 NUC=NUC*OR(OPZ)
7120 RETURN
7200 RESTORE 10100:RETURN
7210 RESTORE 10160:RETURN
7220 RESTORE 10210:RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA "TESTS con ATRIBUTOS (generacion)"
10070 DATA 30,31,28,29,20,16,15,13,25,19,17,0,0,0,0
10080 DATA 3
10090 DATA 1,1,1,10,1,11
10100 DATA "Introducir la sigla del test"
10120 DATA 3,2,0,2,31,11
10160 DATA " ",3,30,0,6,6,10
10210 DATA " "
10220 DATA 1,1,0,21,21,10
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 LOCATE X,Y
11040 RETURN
13000 REM -----
13010 REM AYUDA
13020 REM -----
13050 LOCATE 0,22
13055 PRINT "AYUDA (barra espaciadora para continuar)";
13060 GOSUB 13600
13110 PRINT "FLECHAS: mueven en las 4 direcciones    ";
13120 GOSUB 13600
13130 PRINT "RETURN: alinea y convalida introduccion ";
13140 GOSUB 13600
13150 PRINT "F1: anula y recomieza el programa       ";
13160 GOSUB 13600
13170 PRINT "F2: anula toda la introduccion          ";
13180 GOSUB 13600
13190 PRINT "F3: borra un caracter                   ";
13200 GOSUB 13600
```

```
13210 PRINT "F4: permite la insercion de espacios    ";
13212 GOSUB 13600
13215 PRINT "F5: graba toda la introduccion          ";
13220 GOSUB 13600
13230 PRINT "                         CTRL Y (AYUDA)    ";
13300 RETURN
13600 A$=INKEY$:IF A$= "" THEN 13600
13605 IF A$<> CHR$(32) THEN 13600
13610 LOCATE 0,22
13620 RETURN
```

Atributos-Uso

```
10 REM ------------------
20 REM TEST con atributos
25 REM      U S O
27 REM   Version MSX
30 REM ------------------
32 MAXFILES=2: SCREEN2:COLOR4,15,7:CLEAR 10000
33 OPEN"GRP:"FOROUTPUTAS#1
35 KEY OFF:KEY 1,CHR$(15):KEY 2,CHR$(16):KEY 3,CHR$(17):KEY 4,CHR$(19):KEY 5,CHR$(20):WIDTH 40
40 DIM TF(15),TP(20),LC(20),ND(20),CX(20),CY(20),DX(20),DY(20)
50 DIM B$(20),D$(20),VO$(10)
55 DIM DT$(10),NO$(100),FO$(100)
60 REM CONSTANTES
70 A1=1:A2=31:BP$="O5L30DFBR8BB":NS$="=":AS$="-":PD$=".":BL$=" "
75 F$=CHR$(1)+CHR$(87)
77 O$="":FOR I=1 TO 32:O$=O$+F$: NEXT I
80 R=RND(-TIME)
100 US$=CHR$(13):ZE$=CHR$(48):GIU$=CHR$(32)
110 HE$="CTRL Y (AYUDA)"
115 H1$=" E.G.S.   Ediciones Forum, S.A."+O$
120 REM CARGA VOCES
130 RESTORE
140 READ PROG$
190 FOR I=1 TO 15
200 READ TF(I)
210 NEXT I
215 REM
216 REM ----
220 REM MAIN
230 REM ----
235 REM
240 OPZ=0:GOSUB 2000
250 X=3:Y=10:K=6:GOSUB11100:PRINT#1,"Posicionar la cinta datos":Y=12:GOSUB11000:PRINT#1,"      y
apretar PLAY."
260 X=4:Y=20:K=1:GOSUB11100:PRINT#1,"HECHO ? (pulsa una tecla)":PLAYBP$
270 IF INKEY$="" THEN 270
280 GOSUB1000
290 X=7:Y=14:K=4:GOSUB11100:PRINT#1,"E S P E R A R":PLAYBP$
295 FOR I=1 TO 500:NEXT
300 OPEN"CAS:ATT"FORINPUTAS#2
310 INPUT#2,NT$
320 FOR I=1 TO 10:INPUT#2,DT$(I):NEXT:CLOSE#2
330 OPEN"CAS:EL"FORINPUTAS#2
340 INPUT#2,MX$:MX=VAL(MX$)
350 FORI=1TOMX:INPUT#2,A$
360 NO$(I)=LEFT$(A$,20):FO$(I)=MID$(A$,21,10):NEXTI:CLOSE#2
370 GOSUB1000:X=.5:Y=10:K=6:GOSUB11100
380 PRINT#1,"Esta cinta contiene el test:":X=11:Y=13:K=1:GOSUB11100:PRINT#1,"** ";NT$;" **"
390 X=4:Y=19:K=2:GOSUB11100:PRINT#1,"Deseas ver las descripciones":Y=Y+1.5:GOSUB 11000:PRINT#1,"
de los atributos ? (S o N)"
400 I$=INKEY$:IFI$="N"ORI$="n"THEN450
410 IF I$<>"s" AND I$<>"S" THEN 400
420 GOSUB 1000:COLOR 6
425 X=6:Y=8:GOSUB 11000:PRINT#1,"LISTA ATRIBUTOS:":COLOR 4
430 FOR Y=10 TO 19:X=1:GOSUB 11000:PRINT#1,DT$(Y-9):NEXT Y
440 X=9:Y=22:K=6:GOSUB11100:PRINT#1,"PULSA UNA TECLA"
445 IF INKEY$="" THEN 445
450 GOSUB1000:X=5:Y=9:K=12:GOSUB11100:PRINT#1,"1) Desarrollo por linea":Y=12:GOSUB 11000:PRINT#1
,"2) Desarrollo por columna":Y=15:GOSUB11000:PRINT#1,"9) Fin programa"
460 X=8:Y=20:K=1:GOSUB11100:PRINT#1,"CUAL ? _"
470 HI=VAL(INKEY$):IF HI=9 THEN END ELSE IF HI<>1 AND HI<>2 THEN 470
475 WO=0
480 ON HI GOSUB 9000,9500
500 IF W4=0 THEN A=7:B=8:A$="ERRONEA" ELSE A=7:B=2:A$="CORRECTA"
501 CLS:LINE(47,76)-(207,106),A,BF
503 CIRCLE(127,91),110,B,,,.2
505 CIRCLE(127,91),80,B,,,.2
506 PAINT(127,110),B
510 COLOR1: PRESET(60,88):PRINT#1,"RESPUESTA ";A$
520 IF W4=0 THEN PLAY"O2L2C","O2L2E","O2L2G" ELSE PLAY"O6L16CDEFGABO7C"
530 COLOR 13:PRESET(70,180):PRINT#1,"PULSA UNA TECLA"
540 IF INKEY$="" THEN 540
545 IF W4=0 THEN WO=1:GOTO 480
550 CLS:OPZ=0:GOSUB 2000:GOTO 370
1000 REM CLS parcial
1010 LINE(0,50)-(255,192),15,BF:RETURN
2000 REM ----------------
2010 REM PRESENTA MASCARA
2020 REM ----------------
2030 CLS
2130 L=LEN(PROG$)
2140 FM=(32-L)/2:FM$=SPACE$(FM)
2150 X=1:Y=.5:K=2:GOSUB11100:PRINT#1,H1$;:COLOR4
2215 Y=4:X=0:GOSUB 11000:PRINT#1, FM$;PROG$;FM$
2216 LINE((FM-1)*8,27)-((32-FM)*8,43),6,B
2217 IFOPZ=0THENRETURN
2218 Y=21:X=0:GOSUB 11000:PRINT#1, O$
2219 Y=22:X=18:GOSUB 11000
2220 PRINT#1, HE$
2240 FOR J=1 TO NUC
2245 A$= ""
2270 K=4:X=DX(J):Y=DY(J):GOSUB 11100:PRINT#1, D$(J)
2280 IFTP(J)=4THENX=CX(J):Y=CY(J):K=6:GOSUB11100:PRINT#1,B$(J):GOTO2420
2290 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J) = 2 THEN C$=NS$:GOTO 2310
2300 C$=AS$
2310 A$=STRING$(N1-N2-1,C$)
2320 IF N2=0 THEN A$= A$+C$:GOTO 2350
2330 A$=A$+PD$
2340 FOR I=1 TO N2:A$=A$+C$:NEXT I
2350 X=CX(J):Y=CY(J):K=6:GOSUB 11100
2360 PRINT#1, A$
2380 B$(J)= SPACE$ (N1-N2-1)
2410 B$(J)= B$(J)+RIGHT$ (A$,N2+1)
2420 NEXT J
2430 RETURN
3000 REM ----------
3010 REM DATA ENTRY
3020 REM ----------
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 PLAY BP$ :FS=0:J=NUC:GOTO 3040
3150 IF J< NUC THEN J=J+1:GOTO 3040
3160 PLAY  BP$:FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 PLAY  BP$:FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB 11000:GOTO 3060
3200 PLAY  BP$:FS=0:GOTO 3040
3210 IF FS>8 THEN FS=0:GOTO 3060
3220 IF FS=8 THEN FS=0:GOTO 3100
3230 RETURN
3240 REM -------------------
3250 REM LECTURA DE UN CAMPO
3260 REM -------------------
3270 FS=0
3280 IF TP(NC)=4 THEN RETURN
3300 IF MID$(B$(NC),CC,1)=PD$ THEN X=X+S:CC=CC+S
3310 GOSUB 11000
3320 A$=INKEY$:IF A$= "" THEN  PRINT#1,"█";:FORI=1TO30:NEXT:K=15:GOSUB11100:PRINT#1,"█":K=6:GOSU
B11100:PRINT#1,MID$(B$(NC),CC,1);:FORI=1TO50:NEXT:GOSUB11000:GOTO3320
3330 CA=ASC(A$): IF CA > A2 THEN 3380
3340 FOR I=1 TO 15
3350 IF TF(I)= CA  THEN FS = I: GOSUB 3630: RETURN
3360 NEXT I
3370 GOTO 3320
3380 IF TP(NC)<>1 THEN 3440
3390 IF CA=45 OR CA=46 OR (CA>=48 AND CA<=57) THEN 3410
```

```
3400 PLAY BP$:GOTO 3320
3410 IF CA <> 46 THEN 3440
3420 GOSUB 4270
3430 IF SWP=1 THEN 3400
3440 S=1: GOSUB 3520
3450 IF CC=LC(NC) THEN RETURN
3460 CC=CC+1:X=X+1:GOTO 3300
3520 REM -----------------------
3530 REM TRANSFERENCIA AL BUFFER
3540 REM -----------------------
3550 IF CC=1 THEN SN$= "":GOTO 3570
3560 SN$=LEFT$(B$(NC),CC-1)
3570 IF CC=LC(NC) THEN DS$= "":GOTO 3590
3580 N=LC(NC)-CC:DS$=RIGHT$ (B$(NC),N)
3590 B$(NC)=SN$+A$+DS$
3620 COLOR15:PRINT#1,"█"
3625 COLOR6:GOSUB11000:PRINT#1,A$:RETURN
3630 REM --------
3640 REM COMANDOS
3650 REM --------
3660 IF FS>5 THEN 3690
3680 RETURN
3690 IF FS=9 THEN GOSUB 13000: RETURN
3700 IF FS=8 THEN GOSUB 4080:RETURN
3710 IF FS<8 THEN RETURN
3720 ON FS-9 GOSUB 3740,3910
3730 RETURN
3740 REM ---------
3750 REM INSERCION
3760 REM ---------
3770 IF CC=LC(NC) THEN PLAY BP$:RETURN
3780 IF CC=1 THEN SN$= "":GOTO 3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DS$=RIGHT$ (B$(NC),N)
3810 IF TP(NC)=1 OR TP(NC)=2 THEN C$=BL$:GOTO 3830
3820 C$=BL$
3830 A$=SN$ +C$+DS$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 PRINT#1, B$(NC):S=-1
3890 IF RIGHT$(B$(NC),1)= PD$ THEN 3780
3900 RETURN
3910 REM -------
3920 REM BORRADO
3930 REM -------
3940 IF TP(NC)<>1  THEN 3960
3950 IF MID$.(B$(NC),2,1) = PD$ AND CC=1 THEN RETURN
3960 IF CC<1 THEN PLAY BP$: RETURN
3970 IF CC=LC(NC) THEN DS$= "": GOTO 4000
3980 N=LC(NC) -CC:DS$ =RIGHT$ (B$(NC),N)
3990 IF CC=1 THEN SN$= "": GOTO 4010
4000 SN$=LEFT$  (B$(NC),CC-1)
4010 IF TP(NC) = 1 OR TP(NC)=2 THEN C$=NS$:GOTO 4030
4020 C$=AS$
4030 B$(NC)= SN$ +DS$ +C$
4040 XX=X:YY=Y
4050 X=CX(NC):Y=CY(NC):GOSUB 11000
4060 X=XX:Y=YY
4070 PRINT#1, B$(NC);:S=1:RETURN
4080 REM ---------
4090 REM ALINEADOS
4100 REM ---------
4110 IF LC(NC) =1 THEN RETURN
4120 N1=LC(NC) -CC+1
4130 IF CC=1 THEN B$(NC) = "": GOTO 4150
4140 B$(NC)= LEFT$ (B$(NC),CC-1)
4150 IF TP(NC) =1 OR TP(NC)=2 THEN 4200
4160 FOR I=1 TO N1
4170 B$(NC)=B$(NC)+BL$
4180 NEXT I
4190 GOTO 4240
4200 FOR I=1 TO N1
4210 B$(NC)=BL$ +B$(NC)
4220 NEXT I
4230 IF RIGHT$ (B$(NC),1) =PD$ THEN B$(NC)= BL$ +LEFT$ (B$(NC),LC(NC)-1)
4240 X=CX(NC):Y=CY(NC)
4250 K=15:GOSUB 11100:PRINT#1,STRING$(LC(NC),"█"):K=6:GOSUB11100
4260 PRINT#1, B$(NC):RETURN
4270 REM ----------
4280 REM CTRL.PUNTO
4290 REM ----------
4300 SWP = 0
4310 IF CC=1 OR CC=LC(NC) THEN SWP =1:RETURN
4320 FOR I=1 TO LC(NC)
4330 IF MID$(B$(NC),I,1)= PD$ THEN I = LC(NC): SWP=1: RETURN
4340 NEXT I: RETURN
7000 REM --------------------
7010 REM LECTURA DE LOS DATOS
7020 REM --------------------
7030 RESTORE 10080
7040 READ NV
7050 FOR I = 1 TO NV:READ NUC(I),OR(I)
7060 NEXT I
7070 NUC=NUC(OPZ)
7080 FORK=0TOOR(OPZ)-1
7090 ON OPZ GOSUB 7190,7200
7100 FOR I=1 TO NUC
7110 READ D$(I+K)
7120 NEXT I
7130 FOR I= 1+K*NUC TO NUC+K*NUC
7140 READ TP(I),LC(I),ND(I),DX(I),CX(I),CY(I)
7150 CY(I)=CY(I)+K:DY(I)=CY(I)
7160 NEXT I,K
7170 NUC=NUC*OR(OPZ)
7180 RETURN
7190 RESTORE 10100:RETURN
7200 RESTORE 10160:RETURN
9000 REM --------------------
9003 REM DESARROLLO POR LINEA
9005 REM --------------------
9006 R$=""
9007 IF WO=1 THEN 9050
9010 N9=INT(RND(1)*MX)+1
9020 IF NO$(N9)="" THEN 9010
9030 OPZ=1:GOSUB 7000:TP(1)=4:LC(1)=20:D$(1)="Elemento:":DX(1)=1:CX(1)=11:DY(1)=9:CY(1)=9:B$(1)=
NO$(N9)
9040 FOR I=2 TO 11:D$(I)=DT$(I-1):NEXT I
9050 GOSUB 2000
9060 X=.5:Y=7:K=2:GOSUB 11100:PRINT#1,"Introduc.'S' para Si 'N' para No"
9065 LINE(5,64)-(250,64),2
9070 GOSUB3000
9080 IFFS=6THEN9050ELSE IF FS=7THEN370
9090 FORI=1TO10:IFB$(I+1)="s"ORB$(I+1)="S"THENR$=R$+"1"ELSER$=R$+"0"
9100 NEXT
9110 IFR$<>FO$(N9)THENW4=0ELSEW4=1
9150 RETURN
9500 REM ----------------------
9503 REM DESARROLLO POR COLUMNA
9505 REM ----------------------
9506 R$="":IFWO=1THEN9570
9507 RO$=""
9510 N9=INT(RND(1)*10)+1
9520 IFDT$(N9)=""THEN9510
9525 OPZ=2:GOSUB7000:TP(1)=4:LC(1)=30:D$(1)="Atributo:":DX(1)=10:CX(1)=1:DY(1)=7.7:CY(1)=9:B$(1)
=DT$(N9)
9530 FORKK=1TO10
9540 N8=INT(RND(1)*MX)+1:IFNO$(N8)=""THEN9540
9550 D$(KK+1)=NO$(N8):RO$=RO$+MID$(FO$(N8),N9,1)
9560 NEXTKK
9570 GOSUB2000
9580 X=.5:Y=6.3:K=2:GOSUB11100:PRINT#1,"Introduc.'S' para Si 'N' para No"
9590 GOSUB3000
9600 IF FS=6 THEN 9570 ELSE IF FS=7 THEN 370
9610 FOR I=1 TO 10:IFB$(I+1)="s" OR B$(I+1)="S" THEN R$=R$+"1" ELSE R$=R$+"0"
9615 NEXT I
9620 IF R$=RO$ THEN W4=1 ELSE W4=0
9630 RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA "TEST con atributos (uso)
10070 DATA 30,31,28,29,20,16,15,13,25,19,17,0,0,0,0
10080 DATA 2
10090 DATA 1,11,1,11
10100 DATA " ",3,1,0,0.5,30.5,10
10160 DATA " ",3,1,0,5,27,10
```

```
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 PRESET(X*8,Y*8),15
11040 RETURN
11100 REM ----------------
11110 REM POS.CURSOR COLOR
11120 REM ----------------
11130 PRESET(X*8,Y*8),15
11135 COLORK
11140 RETURN
13000 REM -----
13010 REM AYUDA
13020 REM -----
13050 GOSUB 13610
13055 PRINT#1, "Barra espaciad. para continuar ";
13060 GOSUB 13600
13110 PRINT#1, "FLECHAS:mueven en las 4 direcc.";
13140 GOSUB 13600
13150 PRINT#1, "F1: si renuncias a la solucion ";
13160 GOSUB 13600
13170 PRINT#1, "F2: anula toda la introduccion ";
13180 GOSUB 13600
13190 PRINT#1, "F3: borra un caracter          ";
13200 GOSUB 13600
13210 PRINT#1, "F5: convalida la introduccion  ";
13220 GOSUB 13600
13230 PRINT#1, "                    CTRL Y (AYUDA)";
13300 RETURN
13600 A$=INKEY$:IF A$= "" THEN 13600
13605 IF A$<> CHR$(32) THEN 13600
13610 PRESET(3,176):COLOR15
13620 PRINT#1, "[redacted]";:COLOR4:PRESET(3,176)
13640 RETURN
```

Símbolos-Generación

```
10  REM ---------------------
20  REM T E S T  con SIMBOLOS
30  REM       generacion
35  REM      Version  MSX
40  REM ---------------------
50  SCREEN1,3:COLOR4,15,7
60  CLEAR14000
70  KEY OFF:KEY5,CHR$(20):WIDTH 32
80  M3=30:REM M3=Max Test
90  DIM SS$(6,M3),COL$(6,M3),RR$(M3)
100 BP$="O5L32DFB"
110 F$=CHR$(1)+CHR$(87)
120 O$="":FOR I=0 TO 39:O$=O$+F$: NEXT I
130 H1$=" E.G.S.   Ediciones Forum, S.A. "
140 X=0:Y=1:GOSUB 11000:PRINTH1$
150 X=6:Y=8:GOSUB 11000:PRINT"TEST CON SIMBOLOS"
160 Y=9:GOSUB11000:PRINT"—————————————————"
170 X=2:Y=15:GOSUB 11000:PRINT"el maximo numero previsto "
180 X=6:Y=17:GOSUB 11000:PRINT"de tests es de ";M3
190 X=8:Y=22:GOSUB 11000:PRINT"pulsa una tecla":PLAYBP$
200 IF INKEY$=""THEN 200
1000 REM -------
1010 REM M A I N
1020 REM -------
1030 REM
1040 RESTORE15000:A$="":FOR I=1 TO 32:READA:A$=A$+CHR$(A):NEXT
1050 SPRITE$(3)=A$
1060 FOR JQ=1 TO M3
1070 GOSUB 13000:REM sprites
1080 SCREEN1
1090 X=0:Y=0:GOSUB 11000:PRINT H1$
1100 GOSUB 1420
1110 FOR I=1 TO 6:IF I=3 THEN NEXT I
1120 SPRITE$(I)=SS$(I,JQ):NEXT
1130 PUTSPRITE1,(19,27),VAL(COL$(1,JQ)),1
1140 PUTSPRITE2,(99,27),VAL(COL$(2,JQ)),2
1150 PUTSPRITE3,(179,27),0,3
1160 PUTSPRITE4,(19,90),VAL(COL$(4,JQ)),4
1170 PUTSPRITE5,(99,90),VAL(COL$(5,JQ)),5
1180 PUTSPRITE6,(179,90),VAL(COL$(6,JQ)),6
1190 X=1:Y=21:GOSUB 11000:PRINT"*Test";STR$(JQ);"* INDICAR LA RESPUESTA";
1200 X=10:Y=23:GOSUB11000:PRINT"<1>,<2>o<3>";
1210 I=VAL(INKEY$):IF I<1ORI>3 THEN VPOKE6927,0:FOR T=1 TO 200:NEXT T:VPOKE6927,1:FOR T=1 TO 200
:NEXT T:GOTO 1210
1220 GOSUB 1230:GOTO 1320
1230 ON I GOSUB 1570,1580,1590
1240 IF SW=3 THEN S=S*(-1):SWAPX1,X2
1250 FOR X=X1 TO X2 STEPS
1260 ON I GOSUB 1510,1530,1550
1270 IF I=3 THEN SWAPX,Y
1280 PUTSPRITEI+3,(X,Y),VAL(COL$(I+3,JQ)),I+3
1290 IF I=3 THEN SWAPX,Y
1300 NEXT:PUTSPRITE3,(0,209),3,3
1310 RETURN
1320 FOR Y=21 TO 23:X=0:GOSUB 11000:PRINT SPACE$(31);:NEXT
1330 X=0:Y=22:GOSUB11000:PRINT"Respuesta";I;" Seguro? (S o N)";
1340 R$=INKEY$:IFR$="S"ORR$="s"THEN1370
1350 IF R$<>"n" AND R$<>"N" THEN 1340
1360 SW=3:X=0:Y=22:GOSUB11000:PRINTSPACE$(33);:GOSUB1230:SW=0:GOTO1150
1370 RR$(JQ)=RIGHT$(STR$(I),1)
1375 SCREEN1:IFJQ=M3THEN2000
1380 X=6:Y=10:GOSUB11000:PRINT"OTRO TEST ? (S o N)"
1390 I$=INKEY$:IFI$="n"ORI$="N"THEN2000
1400 IF I$<>"s" AND I$<>"S" THEN 1390
1410 FOR I=1 TO 500:NEXT:NEXT JQ
1420 REM
1430 FOR K=1 TO 21 STEP10
1440 FOR H=2 TO 10 STEP8
1450 X=K:Y=H:GOSUB 1490
1460 NEXT H,K
1470 X=2:Y=18:GOSUB 11000:PRINT". 1 /     . 2 /     . 3 /"
1480 RETURN
1490 GOSUB 11000:PRINT"┌──────┐":FOR I=1 TO 5:Y=Y+1:GOSUB11000:PRINT"│      │":NEXT I:Y=Y+1:GOSUB
11000: PRINT"└──────┘"
1500 RETURN
1510 REM 1
1520 Y=97.2-X*.39:RETURN
1530 REM 2
1540 Y=167.4-X*.78:RETURN
1550 REM 3
1560 Y=180:RETURN
1570 X1=20:X2=180:S=1:RETURN
1580 X1=100:X2=180:S=1:RETURN
1590 X1=90:X2=27:S=-1:RETURN
2000 REM ------------------
2010 REM ESCRITURA FICHEROS
2020 REM ------------------
2030 CLS:X=2:Y=7:GOSUB11000:PRINT"Posicionar la cinta datos":X=5:Y=10:GOSUB11000:PRINT"y apretar
";"REC y PLAY";:Y=16:GOSUB 11000:PRINT"Despues pulsar una tecla"
2040 IF INKEY$="" THEN 2040
2050 CLS:X=9:Y=12:GOSUB11000:PRINT"E S P E R A R"
2060 A$="":OPEN"CAS:az"FOR OUTPUT AS #1
2070 FOR I=1 TO 6:PRINT #1,A$:NEXTI:CLOSE#1:REM espacio cinta
2080 OPEN"CAS:TESTS"FOR OUTPUT AS #1
2090 A$=STR$(JQ):AA$="*"+A$+SPACE$(9-LEN(A$))
2100 PRINT#1,AA$
2110 FOR I=1 TO JQ
2120 FOR J=1 TO 6:IF J=3 THEN NEXT J
2130 AA$="":BB$=SS$(J,I):FORH=1TO32
2140 IFMID$(BB$,H,1)=CHR$(0)THENAA$=AA$+CHR$(2)+CHR$(2):GOTO2160
2150 AA$=AA$+CHR$(1)+MID$(BB$,H,1)
2160 NEXTH
2170 PRINT #1,AA$
2180 PRINT #1,COL$(J,I):NEXTJ
2190 PRINT#1,RR$(I)
2200 NEXT I
2210 CLOSE#1
2220 CLS:X=8:Y=12:GOSUB11000:PLAYBP$+BF$:PRINT"FIN PROGRAMA"
2230 Y=20:GOSUB 11000:END
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 LOCATE X,Y
11040 RETURN
13000 REM ------------------
13010 REM INTRODUCCION DIBUJO
13020 REM ------------------
13030 SCREEN1:WIDTH32
13040 FORJK=1TO6:IFJK=3THEN13440
13050 SPRITE$(0)=CHR$(0)
```

```
13060 PUTSPRITE0,(200,20),1,0
13070 CLS:COLOR4,15
13080 X=0:Y=0:GOSUB11000:PRINTH1$
13090 X=12:Y=2:GOSUB11000:PRINT"1111111":X=3:Y=3:GOSUB11000:PRINT"1234567890123456"
13100 X=2:Y=4:GOSUB11000:PRINTCHR$(1);CHR$(&H58);LEFT$(O$,32);CHR$(1);CHR$(&H59)
13110 X=0:A=0:FOR Y=5 TO 20:A=A+1:A$=STR$(A):GOSUB 11000:PRINTRIGHT$(A$,2);"|";SPACE$(16);"|";:N
EXTY:X=2:GOSUB 11000:PRINTCHR$(1);CHR$(&H5A);LEFT$(O$,32);CHR$(1);CHR$(&H5B);
13120 :X=20:Y=10:GOSUB11000:PRINT"'E' enciende":Y=11:GOSUB11000:PRINT"'A' apaga";
13130 PLAYBP$:X=21:Y=13:GOSU B11000:PRINT"Las flechas";:Y=14:GOSUB 11000:PRINT"desplazan";:Y=15:
GOSUB11000:PRINT"el cursor.";:X=20:Y=18:GOSUB11000:PRINT"F5 confirma"
13140 X=6:Y=22:GOSUB11000:PRINT"Figura n.";JK;
13150 Q=1:W=1:P=0
13160 A$=INKEY$
13170 IFA$<>""THEN13210
13180 LOCATEQ+3,W+5,1:FORT=1TO30:NEXT
13190 LOCATEQ+3,W+5,0:FORT=1TO30:NEXT
13200 GOTO13160
13210 IFA$=CHR$(28)THENQ=Q+1:IFQ>15THENQ=15
13220 IFA$=CHR$(29)THENQ=Q-1:IFQ<0THENQ=0
13230 IFA$=CHR$(30)THENW=W-1:IFW<0THENW=0
13240 IFA$=CHR$(31)THENW=W+1:IFW>15THENW=15
13250 IFQ>7THENP=1ELSEP=0
13260 IF A$="e" OR A$="E" THEN GOSUB13460:LOCATEQ+3,W+5:PRINT"█"
13270 IF A$="a" OR A$="A" THEN GOSUB 13490:LOCATE Q+3,W+5:PRINT" "
13280 IFA$=CHR$(20)THEN13300
13290 GOTO 13160
13300 X=20:FOR Y=10 TO 13:GOSUB 11000:PRINTSPACE$(12);:NEXTY
13310 PLAYBP$:X=20:Y=13:GOSUB 11000:PRINT"Una tecla";:Y=14:GOSUB 11000:PRINT"para elegir";:Y=15:
GOSUB 11000:PRINT"el color de";:Y=16:GOSUB 11000:PRINT"la figura":Y=18:GOSUB 11000:PRINT"F5 CONF
IRMA"
13320 A=1
13330 A$=INKEY$:IFA$=""THEN13330
13340 IFA$=CHR$(20)THENPLAYBP$:GOTO13380
13350 A=A+1:IFA>14THENA=1
13360 PUTSPRITE0,(200,20),A,0
13370 GOTO 13330
13380 REM
13390 S$="":FOR I=BASE(9) TO BASE(9)+31
13400 S$=S$+CHR$(VPEEK(I))
13410 NEXT I
13420 SS$(JK,JQ)=S$
13430 COL$(JK,JQ)=STR$(A)
13440 NEXTJK
13450 RETURN
13460 REM
13470 S=BASE(9)+16*P+W:VPOKES,VPEEK(S)OR2^(7-Q+8*P)
13480 RETURN
13490 REM
13500 S=BASE(9)+16*P+W:VPOKES,VPEEK(S)ANDNOT(2^(7-Q+8*P))
13510 RETURN
15000 DATA0,7,8,16,16,16,0,1,2,2,2,2,2,0,2,2,0,0,128,64,64,64,128,0,0,0,0,0,0,0,0
```

Símbolos-Uso

```
10   REM ----------------------
20   REM T E S T  con SIMBOLOS
30   REM        U S O
35   REM    Version   MSX
40   REM ----------------------
50   SCREEN1,3 :COLOR4,15,7
60   CLEAR14000
70   KEY OFF:KEY5,CHR$(20):WIDTH 32
80   M3=30:REM M3=Max Test
90   DIM SS$(6,M3),COL$(6,M3),RR$(M3)
100  BP$="O5L32DFB"
110  F$=CHR$(1)+CHR$(87)
120  O$="":FOR I=0 TO 39:O$=O$+F$: NEXT I
130  H1$=" E.G.S.    Editorial Forum,S.A."
140  GOSUB 150:FF=1:GOSUB 1600
150  X=0:Y=1:GOSUB 11000:PRINTH1$
160  X=4:Y=5:GOSUB 11000:PRINTCHR$(1);CHR$(&H44);"────────────────────";CHR$(1);CHR$(&H44)
170  Y=6:GOSUB11000:PRINT"|TEST CON SIMBOLOS (uso)|"
180  Y=7:GOSUB 11000:PRINTCHR$(1);CHR$(&H44);"────────────────────";CHR$(1);CHR$(&H44)
185  IF FF=0 THEN RETURN
190  X=2:Y=12:GOSUB 11000:PRINT"El numero de tests contenidos"
200  X=4:Y=14:GOSUB 11000:PRINT"en esta cinta  es ";MX;"."
210  X=8:Y=22:GOSUB 11000:PRINT"pulsa una tecla":PLAYBP$
220  IFINKEY$=""THEN220
1000 REM -------
1010 REM M A I N
1020 REM -------
1030 REM
1040 RESTORE12000:A$="":FOR I=1 TO 32:READA:A$=A$+CHR$(A):NEXT
1050 SPRITE$(3)=A$
1060 FOR JQ=1 TO MX
1070 SCREEN1
1080 X=0:Y=0:GOSUB 11000:PRINTH1$
1090 GOSUB 1420
1100 FOR I=1 TO 6:IF I=3 THEN NEXT I
1110 SPRITE$(I)=SS$(I,JQ):NEXT
1120 PUTSPRITE1,(19,27),VAL(COL$(1,JQ)),1
1130 PUTSPRITE2,(99,27),VAL(COL$(2,JQ)),2
1140 PUTSPRITE3,(179,27),0,3
1150 PUTSPRITE4,(19,90),VAL(COL$(4,JQ)),4
1160 PUTSPRITE5,(99,90),VAL(COL$(5,JQ)),5
1170 PUTSPRITE6,(179,90),VAL(COL$(6,JQ)),6
1180 X=1:Y=21:GOSUB11000:PRINT"*Test";STR$(JQ);"* INDICAR LA RESPUESTA:";
1190 X=10:Y=23:GOSUB11000:PRINT"<1>,<2>o<3>";
1200 I=VAL(INKEY$):IF I<1ORI>3 THEN VPOKE6927,0:FOR T=1 TO 200:NEXT T:VPOKE6927,1:FOR T=1 TO 200
:NEXT T:GOTO 1200
1210 GOSUB 1220:GOTO 1310
1220 ON I GOSUB 1570,1580,1590
1230 IF SW=3 THEN S=S*(-1):SWAPX1,X2
1235 GOSUB 30000
1240 FOR X=X1 TO X2 STEPS
1250 ON I GOSUB 1510,1530,1550
1260 IF I=3 THEN SWAPX,Y
1270 PUTSPRITEI+3,(X,Y),VAL(COL$(I+3,JQ)),I+3
1280 IF I=3 THEN SWAPX,Y
1290 NEXT:PUTSPRITE3,(0,209),3,3
1300 RETURN
1310 FOR Y=21 TO 23:X=0:GOSUB 11000:PRINT SPACE$(31);:NEXT
1320 X=2:Y=22:GOSUB 11000:PRINT"Respuesta";I;". Seguro ? (S o N)";
1330 R$=INKEY$:IF R$="S" OR R$="s" THEN 1360
1340 IF R$<>"n" AND R$<>"N" THEN 1330
1350 SW=3:X=2:Y=22:GOSUB 11000:PRINT SPACE$(31);:GOSUB 1220:SW=0:GOTO1140
1360 IFI<>VAL(RR$(JQ))THENBEEP:PLAY"O2L2V15C","O2L2V15E","O2L2V15G":G4=1:GOTO1350
1370 PLAY"O5L32CDEFGABO6CR16CO5BAGFEDCR16ECR64EC":FOR J=1 TO 3:IF J=I THEN 1377
1375 VPOKE6927+4*J,0
1377 NEXT J:FOR I=1 TO 1000:NEXT
1380 SCREEN1:X=6:Y=10:GOSUB 11000:PRINT"OTRO TEST ? (S o N)"
1390 I$=INKEY$:IFI$="n"ORI$="N"THEN1412
1400 IF I$<>"s" AND I$<>"S" THEN 1390
1410 NEXT JQ
1412 CLS:X=9:Y=12:GOSUB 11000:PRINT"FIN PROGRAMA":Y=20:GOSUB 11000:END
1420 REM
1430 FOR K=1 TO 21 STEP10
1440 FOR H=2 TO 10 STEP8
1450 X=K:Y=H:GOSUB1490
1460 NEXTH,K
1470 X=2:Y=18:GOSUB11000:PRINT". 1 /     . 2 /     . 3 /"
1480 RETURN
1490 GOSUB 11000:PRINT"┌─────┐":FOR I=1 TO 5:Y=Y+1:GOSUB 11000:PRINT"|     |":NEXTI:Y=Y+1:GOSUB
11000:PRINT"└─────┘"
1500 RETURN
1510 REM 1
1520 Y=97.2-X*.39:RETURN
1530 REM 2
1540 Y=167.4-X*.78:RETURN
1550 REM 3
1560 Y=180:RETURN
1570 X1=20:X2=180:S=1:RETURN
1580 X1=100:X2=180:S=1:RETURN
1590 X1=90:X2=27:S=-1:RETURN
1600 REM ---------------
1610 REM LECTURA FICHERO
1620 REM ---------------
1630 X=3:Y=10:GOSUB11000:PRINT"Posicionar la cinta datos":X=8:Y=12:GOSUB 11000:PRINT"y apretar
PLAY.";:X=5:Y=16:GOSUB 11000:PRINT"Despues pulsar una tecla"
1640 IFINKEY$=""THEN1640
1650 CLS:X=9:Y=12:GOSUB 11000:PRINT"E S P E R A R"
1660 OPEN"CAS:TESTS"FOR INPUT AS #1
1670 INPUT#1,AA$
1680 MX=VAL(MID$(AA$,2,9))
1690 FOR I=1 TO MX
1700 FOR J=1 TO 6:IF J=3 THEN NEXT J
1710 INPUT #1,AA$
1720 BB$="":FOR H=1 TO 64 STEP2
```

```
1730 IF MID$(AA$,H,1)=CHR$(2) THEN BB$=BB$+CHR$(0):GOTO 1750
1740 BB$=BB$+MID$(AA$,H+1,1)
1750 NEXT H
1760 SS$(J,I)=BB$
1770 INPUT #1,COL$(J,I):NEXT J
1780 INPUT#1,RR$(I)
1790 NEXTI
1800 CLOSE#1
1810 RETURN
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 LOCATE X,Y
11040 RETURN
12000 DATA0,7,8,16,16,16,0,1,2,2,2,2,2,0,2,2,0,0,128,64,64,64,128,0,0,0,0,0,0,0,0
30000 REM
30005 IF G4=1 THEN G4=0:RETURN
30010 BEEP:F1=2:F2=8:F3=1
30020 IF I=2 THENF 1=3:F2=6
30030 IF I=3 THENF 1=4:F2=6
30040 IF SW=3 THEN SWAPF1,F2:F3=-1
30050 FOR FI=F1 TO F2 STEPF3
30060 A$="O"+STR$(FI)
30070 C$="L10S8M550"+A$+"CDEFG"
30080 D$="L64S8M550"+A$+"CEG"
30090 PLAYC$,D$
30100 NEXT
30110 RETURN
```

RELACION DE LAS VARIABLES - PHILIPS VG 8010

Puntuación - Generación

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en pantalla del campo de introducción
DX(*) **DY(*)**	= coordenadas en pantalla de descripción
TF(*)	= matriz numérica con códigos ASCII de teclas comando
D$(*)	= descripciones en las máscaras video
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de intr. alfanuméricos
NS$	= carácter "#"; identifica los campos de intr. numéricos
NUC	= número campos de intr. en la máscara video activada
MX	= número preguntas
(*) **TS$(*)**	= contiene la pregunta del test
(*) **RI$(*)**	= contiene las respuestas y la correspondiente puntuación
H1$	= cabecera
PROG$	= nombre del programa

(*) En la segunda fase del programa, TS$(*) y RI$(*) contienen respectivamente las gamas de puntuación y los resultados del test

Puntuación - Uso

MX	= número preguntas
TS$(*)	= contiene la pregunta del test
(*) **RI$(*)**	= contiene las respuestas y la correspondiente puntuación
PP	= totalizador de los puntos obtenidos
MI(*) **MX(*)**	= contienen las gamas de puntuación (MI(*) = mínimo; MX(*) = máximo)
H1$	= cabecera
PROG$	= nombre del programa

(*) En la segunda fase del programa, RI$(*) contiene los resultados del test

Abributos - Generación

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en pantalla del campo de introducción
DX(*) **DY(*)**	= coordenadas en pantalla de descripción
TF(*)	= matriz numérica con códigos ASCII de teclas comando
D$(*)	= descripciones en las máscaras video
VO$(*)	= voces del menú principal
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de intr. alfanuméricos
NS$	= carácter "#"; identifica los campos de intr. numéricos
NUC	= número campos de intr. en la máscara vídeo activada
ATS(*)	= contiene los atributos
ELS(*)	= contiene los elementos
M3	= máximo número de elementos
H1$	= cabecera
PROG$	= nombre del programa

NT$	= nombre del test

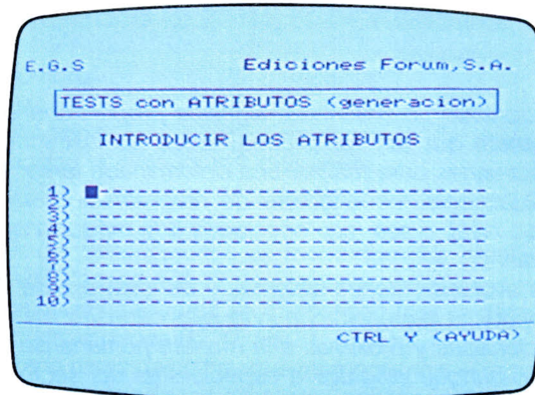
Atributos - Uso

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*) **CY(*)**	= coordenadas en pantalla del campo de introducción
DX(*) **DY(*)**	= coordenadas en pantalla de descripción
TF(*)	= matriz numérica con códigos ASCII de teclas comando
D$(*)	= descripciones en las máscaras vídeo
BL$	= blank
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de intr. alfanuméricos
NS$	= carácter "#"; identifica los campos de intr. numéricos
NUC	= número campos de intr. en la máscara vídeo activada
NT$	= nombre del test
DT$(*)	= contienen los atributos
MX	= número de elementos
NO$(*)	= contiene elementos
FO$(*)	= contiene los flags de cada elemento
W4	= flag respuesta (W4 = 0 resp. errónea; W4 = 1 resp. correcta)

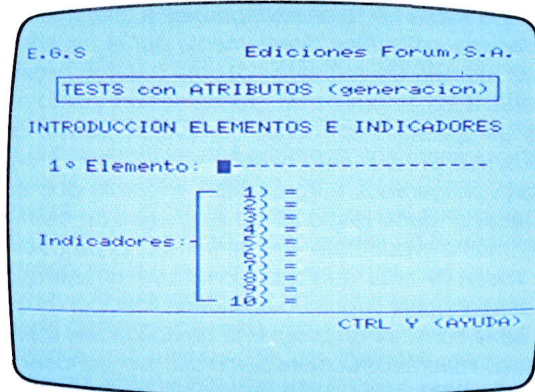
Símbolo - Generación y Uso

H1$	= cabecera
M3	= máx. número de tests
SS$ (6,M3)	= contiene los sprites
COL$ (6,M3)	= contiene en número del color de los sprites
RR$(M3)	= contiene las respuestas correctas
MX	= número de tests (en el programa Uso)

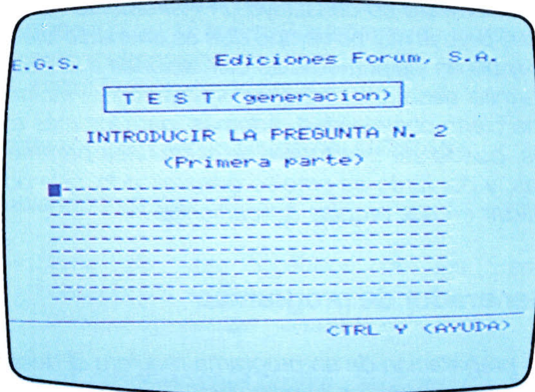
El desarrollo del procedimiento es análogo a los de las otras versiones. También en este caso, aparte de elementos marginales como los colores, la máscara es idéntica para todas las versiones.



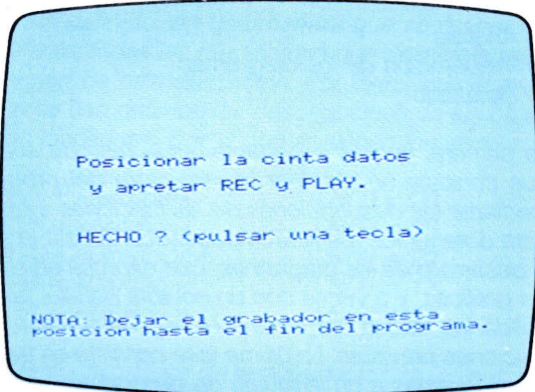
En este caso, la facilidad de incluir en la misma pantalla vídeo tanto texto como gráficos, permite estructurar de manera diferente algunas máscaras. La foto muestra la fase de asociación entre los atributos y los elementos.



Test con puntuación. La foto muestra la máscara vídeo utilizada para la introducción de una pregunta. Por motivos de espacio en la pantalla, el texto se divide en dos presentaciones sucesivas.



La versión MSX está orientada al uso con grabador, por lo que en todos los programas se incluyen las funciones de aviso para el usuario de que debe predisponer la unidad de grabación.



El ordenador para el ordenador

Programar un ordenador significa proporcionarle, de manera que las comprenda, todas las instrucciones necesarias para realizar una determinada tarea. Inicialmente, los lenguajes de programación eran de bajo nivel, o sea más orientados a la máquina que al usuario. La evolución de los sistemas y del software de base condujo a la creación de lenguajes cada vez más evolucionados y próximos a la manera humana de razonar, aunque dirigidos a especialistas del sector. Recientemente, han aparecido programas de aplicación de tipo «abierto», o sea adaptables a diferentes necesidades y utilizables directamente por el usuario final. Esta «familia de software» deberá sustituir necesariamente a los lenguajes de programación, por lo menos en las aplicaciones menos complejas, puesto que el usuario final no puede ni debe estar ligado a programas configurados e inmutables, mientras que el programador debe disponer de lenguajes de nivel cada vez más elevado para ahorrar tiempo. El procedimiento ilustrado más adelante constituye un ejemplo de macrolenguaje orientado al usuario final, con el que es posible construir un programa de aplicación utilizando pocas macroinstrucciones, incluso sin conocer nada de programación. Este software no constituye un instrumento «profesional», pero al mismo tiempo que es un medio didáctico muy útil en las primeras aproximaciones a la máquina, permite desarrollar muchas aplicaciones, incluso de una cierta complejidad. Además, para los más expertos, puede ser una fuente de subrutinas parametrizadas, y por tanto de empleo generalizado, que podrán utilizar en sus propias aplicaciones.

Generador de programas

La preparación de un programa requiere el desarrollo de tres actividades principales:

1 / Análisis
2 / Preparación de los diagramas
3 / Detallado

La primera, la más importante, es la fase de análisis, que consiste en plantear la estructura del programa mediante las descripciones de las funciones a realizar para obtener los resultados deseados. A ésta le sigue el desarrollo de los diagramas, que expresa en símbolos gráficos, y a veces con un notable detalle, las modalidades según las cuales deben desarrollarse las funciones previstas. La última fase consiste en traducir los diagramas a un lenguaje de programación.
En términos de utilización de recursos, esta última fase es la más laboriosa, y también la que puede producir más fácilmente errores y dificultades de detallado del software.
Estos aspectos negativos se deben a la necesidad, en esta fase, de utilizar un lenguaje más próximo al ordenador que al hombre, por lo que un simple concepto, como por ejemplo «Busca en el archivo el registro que en el campo ciudades tiene el valor Bilbao», debe traducirse en una serie de funciones más o menos elementales según el lenguaje utilizado. La forma de esta traducción depende mucho del programador. Incluso utilizando el mismo lenguaje, personas distintas usarán métodos diferentes para obtener la misma finalidad. Esta situación conduce a dos inconvenientes: posibilidad de errores y dificultades en las modificaciones, si éstas deben ser realizadas por otro programador que no sea el que ha escrito la primera versión.
Un método para reducir estos inconvenientes ya se ha presentado, y esta obra lo emplea profusamente. En general, consiste en utilizar módulos de empleo generalizado que puedan realizar las funciones más recurrentes en los programas de aplicación. Un ejemplo inmediato es la data entry. Con esta subrutina es posible la gestión completa del video, simplemente definiendo algunos parámetros (tipo del campo, longitud, posición, etc.).
Esta forma de parametrización constituye ya de por sí una primera aproximación a la preparación de un «generador de programas».
Sin embargo, tiene dos defectos: requiere el conocimiento del lenguaje de programación utilizado (Basic) y necesita modificaciones en el interior del programa, por ejemplo en los DATA y en los flags.
En cambio, el generador de programas debe ser un software que pueda gestionar cualquier aplicación sin necesidad de modificaciones, sino sólo mediante algunas instrucciones. (Obsérvese que el término «generador de programas» se utiliza inadecuadamente, puesto que se trata de un generador de aplicaciones, porque su empleo no produce un listado, o sea otro programa, sino el desarrollo de la aplicación).
Analizando las funciones necesarias en la mayoría de programas, es evidente que éstas pueden dividirse en unos pocos grupos principales, como:

- Entrada y salida de datos
- Desarrollo de los cálculos
- Búsqueda de un dato en un fichero
- Desarrollo de bucles
- Decisiones

Para cada una de ellas existen instrucciones, por ejemplo en Basic, que pueden desarrollarlas, pero son instrucciones necesariamente muy generales, y cada

una activa funciones más elementales de lo que necesita el usuario del software. Por ejemplo, la instrucción INPUT sirve para adjudicar un valor a una variable, o sea es una instrucción de entrada; pero con ella no puede gestionarse una máscara video ni cualquier otra forma de introducción de datos. Efectivamente, una introducción de datos tiene como consecuencia, por lo menos, la necesidad de validar el valor introducido y de permitir correcciones, posibilidades que no están previstas en la instrucción INPUT, mientras que se convierten en automáticas utilizando la data entry. Análogamente, las otras funciones pueden parametrizarse para dar lugar a una serie de rutinas de empleo generalizado. Con estas rutinas es posible construir una forma de lenguaje de alto nivel y de naturaleza descriptiva.

Por ejemplo, la frase anterior «Busca en el archivo...», que expresa una necesidad del usuario, puede ser traducida para su ejecución desarrollando todas las instrucciones Basic necesarias no parametrizadas, o sea utilizando un archivo bien determinado, un campo definido y, como elemento de selección, la cadena «Bilbao».

Es evidente que esta rutina tiene un empleo circunscrito a esta aplicación particular; en este caso, ningún programador utilizaría un método como éste, porque por lo menos el elemento de selección debería parametrizarse, o sea utilizar una cadena general en la que introducir el nombre a buscar. De manera análoga bastaría con parametrizar las otras magnitudes, como el nombre del fichero en el que realizar la búsqueda y la posición del campo. De esta manera se dispondría de una rutina que podría buscar un valor particular en un campo cualquiera. Por tanto, se trata de introducir los parámetros «desde fuera».

La generalización indicada no presenta ninguna dificultad si se obtiene a nivel de programación. Efectivamente, el nombre del fichero y el valor del campo («Bilbao») pueden parametrizarse fácilmente utilizando dos variables cadena en las que cargar cada vez los valores; en lo que respecta a la posición del campo en el interior del registro, la parametrización es mucho menos sencilla, puesto que debe utilizarse un puntero que debe implantarse al valor oportuno antes de utilizar la rutina. Esto puede conseguirse fácilmente modificando el programa, pero a cambio de una complicación para el usuario, que es precisamente lo que se pretende evitar con este software.

Para analizar la lógica seguida en la implantación del programa «generador de programas» es conveniente referirse a un ejemplo concreto. Supongamos que se quiere formar un listín de direcciones. Básicamente se trata de estructurar una máscara video que contenga todos los campos previstos (Nombre, Apellido, etc.) y una serie de subrutinas que actualicen el fichero o realicen selecciones. En realidad, definiendo la máscara video también se definen las posiciones de los distintos campos en el interior del registro.

Por ejemplo, suponiendo que la máscara sea:

Nombre	*20 caracteres*
Apellido	*30 caracteres*
Dirección	*20 caracteres*
Ciudad	*12 caracteres*
Teléfono	*11 caracteres*

En las descripciones se memorizan los nombres de cada campo (D$(1) = «Nombre», etc.) y los valores asumidos cada vez se memorizan (por la data entry) en las cadenas B$(1), B$(2), etcétera.

Como estos datos hay que memorizarlos, el registro se forma sumando las diferentes cadenas B$(I), con I = = 1 ÷ 5. Por tanto, el registro será:

B$(1) + B$(2) + B$(3) + B$(4) + B$(5)

y tendrá una longitud fija, puesto que si un campo no se utiliza, se llena con espacios en blanco; en consecuencia, en cada registro, en el byte 21 empezará el Apellido, en el 51 la Dirección, y así sucesivamente. En general, el campo «N» empieza en la posición correspondiente a la suma de las longitudes de los campos que lo preceden + 1. Por ejemplo, para determinar de dónde debe extraerse el campo Ciudad, deben sumarse los valores de los campos precedentes (20 + 30 + 20 = 70) y al resultado sumarle 1, por lo que el campo Ciudad empezará en el byte 71 y tendrá una longitud de 12 bytes. Por tanto, para extraerlo del registro, y suponiendo que se ha leído en la variable A$, basta la instrucción MID$:

Contenido del campo Ciudad = MID$(A$,K,L)

Habiendo indicado:

K = Suma de las longitudes de los campos precedentes + 1
L = Longitud del campo «Ciudad»

De estas observaciones es inmediato deducir un método de parametrización «externo». Supongamos que se han definido, de una manera que se ilustrará más adelante, todas las magnitudes que definen la máscara video de esta aplicación. Por tanto, con ellas también se han definido las descripciones de los campos y las longitudes, por lo que se tienen los valores:

D$(1) = «Nombre»	*LN(1) = 20*
D$(2) = «Apellido»	*LN(2) = 30*
D$(3) = «Dirección»	*LN(3) = 20*
D$(4) = «Ciudad»	*LN(4) = 12*
D$(5) = «Teléfono»	*LN(5) = 11*

Por tanto, en la fase de lectura del fichero es suficiente con indicar qué campo se desea; por ejemplo, para la «Ciudad», es el número 4.

La subrutina generalizada deberá prever un bucle de 1 a 3 (número del campo seleccionado – 1) que sume las longitudes (LN(*)) de los campos que lo preceden, obteniendo así el valor de K, que debe utilizarse en la instrucción MID$. El otro parámetro es el valor LN(4), suministrado directamente en los parámetros. Aplicando estos conceptos se obtiene una rutina que puede escribir o extraer un campo de cualquier fichero, naturalmente definido anteriormente. El campo así extraído puede transferirse a una matriz dedicada, con la que se realizará la comparación con el valor a buscar. De esta manera se obtiene una serie de rutinas parametrizadas que realizan la búsqueda de un determinado valor en los registros de un fichero. Los parámetros a implantar son:

1 / Nombre del fichero
2 / Número del campo
3 / Valor a buscar

Tal y como se han descrito las funciones, aún no están en la forma definitiva, porque obligan al usuario a recordar la posición en la que ha definido cada campo. Por tanto, hace falta otra subrutina que descargue al usuario de esta tarea.
Una vez introducidas las descripciones de los distintos campos en las variables D$(*), no existe ninguna razón para que el usuario deba recordar sus posiciones, puesto que es más sencillo utilizar una subrutina que las determine. En el ejemplo anterior hay que determinar la posición del campo «Ciudad», lo cual puede hacerse introduciendo la palabra «Ciudad» en cualquier cadena y comparándola con el contenido de D$(*). La comparación tendrá resultado positivo con D$(4), y el índice de valor 4 será precisamente el número buscado.
Es decir, la parametrización completa de la subrutina de búsqueda se obtiene previendo un bucle que seleccione el campo particular entre los definidos. De esta manera ya no es necesario implantar flags a nivel de programación, sino que todos los parámetros pueden definirse mediante un coloquio, o sea son modificables por el usuario final.
Queda todavía un punto por aclarar. En la data entry normal, las características de los campos: Nombres (D$(*)), Longitudes, etc., se asignan utilizando unas DATA, por lo que no son modificables por el usuario. Para obtener esta nueva funcionalidad hay que prever una data entry que permita la introducción de los valores por la data entry. Esta lógica, que parece un juego de palabras, en realidad sólo significa que los valores para estructurar la data entry (descripciones, longitudes, etc.), al principio pueden ser considerados, a su vez, como datos y, por tanto, pueden ser leídos por una data entry. Esta data entry inicial, después de adquirir los valores en los buffers B$(*) usuales, los transfiere a D$(*), LN(*). etc., evitando así la necesidad de las intrucciones DATA.

En este punto, un software estructurado así ya permite utilizar muchas funciones por introducción directa de los parámetros, sin ninguna modificación de los programas. Por tanto, supondremos que se dispone de una subrutina que, utilizando la data entry normal, atribuye los valores a las variables D$(*), LN(*), etc., que llamaremos «subrutinas de las definiciones».
Para realizar la búsqueda de un determinado valor hace falta una segunda rutina que, dada la descripción, pueda determinar el byte de principio del campo y compararlo con el valor. Esta rutina, que a nivel funcional se reduce a un bucle para la búsqueda de la posición y a una comparación, necesita 3 parámetros: Nombre del fichero, Nombre del campo y Valor a utilizar como término de comparación. El nombre del fichero puede darse en la subrutina de definición, por lo que sólo son necesarios la descripción y el valor.
Estas introducciones, que tienen el aspecto de un comando, pueden obtenerse a través de un coloquio, pero esto haría perder generalidad al software. Por tanto, hay que «inventar» frases que, oportunamente decodificadas, proporcionan al ordenador todos los elementos necesarios.
Por ejemplo, la primera fase, que permite identificar las descripciones de los campos y las demás características, puede indicarse con la palabra «DEFINE». El ordenador, al analizar la cadena introducida, reconoce la palabra y pasa el control a la subrutina de las definiciones. De un modo análogo, la búsqueda puede activarse con la palabra «BUSCA».
De esta manera se pueden identificar muchas funciones de base a las que corresponderían otros tantos procedimientos, obteniendo así un lenguaje «por funciones», y no por instrucciones.
A este concepto debe prestársele mucha atención, puesto que es básico para la comprensión del programa. Utilizando un lenguaje de programación normal, cada función debe ser desglosada en las instrucciones simples, y esto también requiere, además del análisis, la preparación de los diagramas y del detallado. Por el contrario, disponiendo de un «lenguaje» que permita describir la función deseada a nivel de síntesis por parte del usuario final, las dos últimas fases ya no son necesarias. De hecho, disponiendo de subrutinas parametrizadas, basta con la relación de las funciones a realizar para obtener el programa; es decir, basta con la fase de análisis, que deberá ser más detallada y descriptiva que en el otro tipo de desarrollo. Esta necesidad no es un defecto ni un aspecto negativo, sino una evolución de metodología. Finalmente, obsérvese que en el texto se ha empleado erróneamente el término «lenguaje» para definir este programa. En realidad no se trata de un verdadero lenguaje, puesto que en él faltan algunos atributos fundamentales para poder ser definido como tal. Sin embargo, es una aplicación muy avanzada y notablemente útil, que además permite una cierta forma de compilación. Es decir, puede utilizarse tanto en interpretado como en compilado.

Uso del programa

Dadas las particularidades del tema, en primer lugar se ilustrarán las características del programa y algunas aplicaciones, dejando el análisis y los diagramas para más adelante. Esto es conveniente debido a las finalidades particulares del software, básicamente orientado a permitir el desarrollo de aplicaciones incluso por personas sin conocimientos específicos.
Anteriormente se han analizado algunas funciones principales, comunes a un amplio conjunto de aplicaciones, y precisamente en esta consideración se basa toda la aplicación.
Como se ha indicado, la primera fase debe ser la de definiciones, o sea la introducción de las características de los diferentes campos de datos que se utilizarán a continuación. Sólo con posterioridad a esta fase es posible desarrollar los procesos previstos. Por tanto, el software debe prever por lo menos dos funciones:

- Definiciones
- Instrucciones

En la primera se introducen todos los nombres simbólicos utlizados y los atributos de los campos asociados, y en la segunda se imparten las intrucciones, siempre a nivel de lenguaje «funcional». Además de las dos funciones principales, deben preverse voces accesorias, como por ejemplo la generación de un listado, por lo que en este caso también se tiene un menú cuya presentación es automática al activar el programa. Las funciones previstas son:

1 / Definiciones: permite introducir nombres y atributos en los campos (en número de registros) del fichero
2 / Instrucciones: en esta fase se definen las funciones a realizar
3 / Listado: permite la presentación en pantalla de las instrucciones introducidas
4 / Ejecución: activa la realización de las instrucciones

El orden en el que se han relacionado es el de prioridad en las aplicaciones. Efectivamente, la primera fase a activar es la 1, y le siguen la introducción y el eventual listado. La última es la ejecución, o sea la 4.

Definiciones. La función está dividida en dos partes: en la primera deben introducirse las características de los campos, o sea:

- Nombre del campo (descripción)
- Tipo (numérico, alfanumérico)
- Número de caracteres previstos (totales)
- Número de caracteres para la parte decimal
- Posición (columna, línea) en la pantalla

El nombre del campo se utiliza para una doble función:

1 / Como descripción en la presentación de las máscaras
2 / Como elemento de identificación en las otras funciones

Por ejemplo, en la constitución de una dirección, uno de los datos a introducir es el nombre de la ciudad, por lo que un campo tiene como descripción la palabra «Ciudad», que también puede utilizarse como referencia para todas las instrucciones previstas en el software. Como se ilustrará más adelante, una vez introducidos los datos, es posible realizar selecciones en base al valor de algunos campos que deben ser direccionados con el nombre dado en la fase de definición. Por tanto, si quisiéramos extraer todas las direcciones de Madrid, la condición que debería producirse es Ciudad = Madrid.
En la primera foto de la página 2352 se ha representado la primera máscara video de las definiciones. Los campos previstos son como máximo 10. El usuario debe introducir su nombre, su tipo (columna Tipo), su longitud (en bytes), el número de sus cifras decimales (columna Dec) y su posición en la pantalla (Columna/ /Línea). Todos los datos deberán ser en lenguaje llano, salvo el tipo, que usa los acostumbrados códigos.
Por ejemplo, para definir el campo Ciudad, de 12 caracteres de longitud y posicionado en la línea 5 a partir de la columna 10, debe introducirse:

Nombre	Tipo	Lon	Dec	Col	Lin
CIUDAD	3	12	0	10	5

Naturalmente, por tratarse de un campo alfanumérico, no se han previsto decimales, por lo que el contenido de la columna Dec es cero (en esta versión debe especificarlo el usuario, pero podría preverlo el programa muy sencillamente en base al código introducido en la columna Tipo).
La segunda fase de las definiciones se utiliza para introducir el nombre del fichero y para seleccionar los campos que deberán memorizarse en él. La primera máscara incluye bien los campos a transferir al fichero, bien los campos que deberán ser utilizados por el procedimiento pero no memorizados. Con esta segunda parte se indican los que constituirán el archivo, o sea se define el registro del fichero de datos. Por ejemplo, si se quiere formar una lista de clientes con el importe progresivo de sus pagos, sólo deberá contener los campos Nombre (alfanumérico) e Importe (numérico). En este último deberán ir sumándose las distintas cantidades a medida que se realizan los pagos, por lo que

la máscara video deberá prever otro campo (numérico) para la introducción del importe correspondiente a cada movimiento simple. Este valor deberá sumarse al progresivo y no se memorizará en el fichero. De ahí la necesidad de indicar, entre todos los campos existentes, cuáles deben quedar excluidos del mismo. En la 2.ª foto de la página 2352 aparece la máscara video correspondiente a esta segunda fase de definición. En ella es necesario indicar:

1 / Nombre del fichero (en esta versión se ha previsto la gestión de un solo fichero)
2 / Longitud en número de registros
3 / Campos, entre los definidos anteriormente, que deben memorizarse.

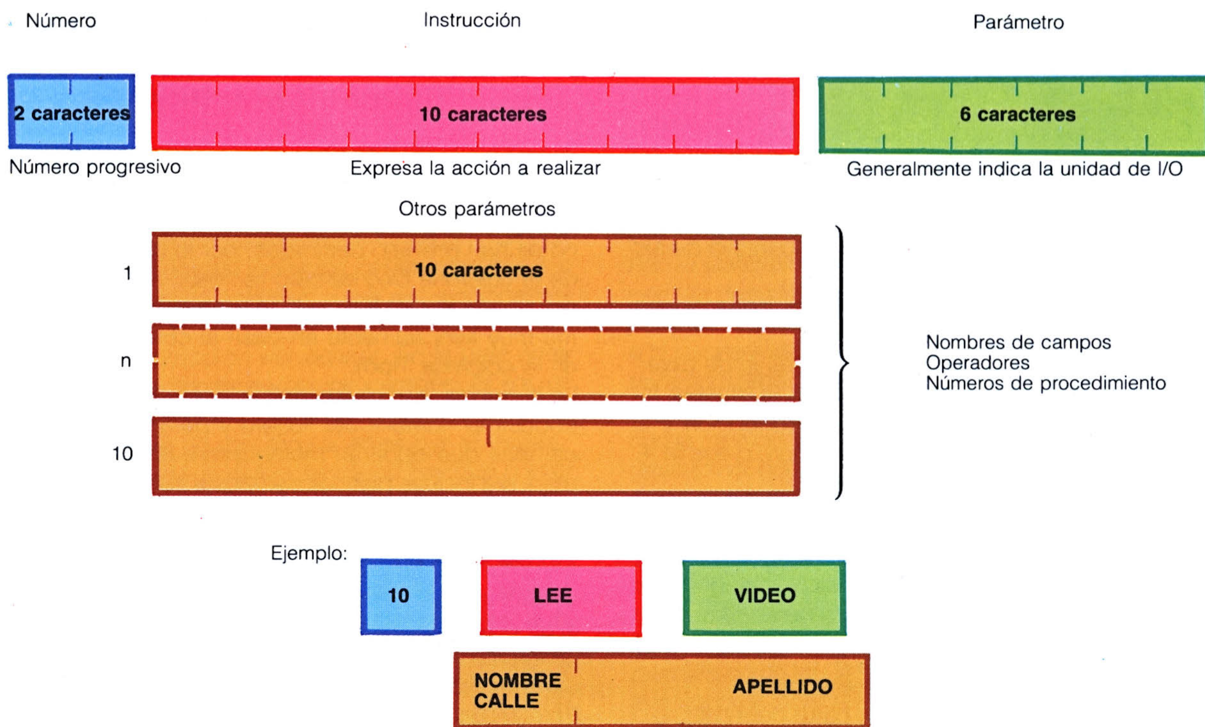
La selección de los campos a memorizar se realiza simplemente introduciendo la letra "S" al lado de cada nombre, cuya presentación, en el mismo orden en que se han introducido durante la primera fase, es automática. Al final del primer procedimiento (Definiciones), articulado en estas dos fases:

- Fase 1: Introducción nombres campos y atributos
- Fase 2: Definición fichero y forma registro

puede pasarse a la escritura del programa propiamente dicho, o sea a la introducción de las instrucciones, que ya no son de nivel elemental como en el Basic, sino que asumen el aspecto de verdaderos procedimientos, o mejor, de «bloques funcionales», cada uno dedicado a una finalidad específica.

Instrucciones. La segunda función a activar, siempre con una máscara video, es la introducción de las instrucciones. En la última foto de la página 2352 se ha representado la máscara que aparece al seleccionar la voz 2 del menú principal.
La estructura general se muestra en la figura de abajo. Los campos existentes son:

- Número: es una numeración progresiva, análoga a la de los números de línea en Basic, que permite identificar cada procedimiento.
- Instrucción: es el nombre del procedimiento que se desea activar entre los previstos (LEE, ESCRIBE, BUSCA, etc.), presentado más adelante.
- Parámetro: generalmente indica la unidad de I/0 hacia la que debe dirigirse el procedimiento; por ejemplo, LEE VIDEO significa que se desea una introducción con data entry. También puede asumir otros significados en función del tipo de instrucción.
- Otros parámetros son los nombres de los campos, operadores (aritméticos y lógicos) o referentes a otros procedimientos que deben utilizarse en el descrito en la primera línea.



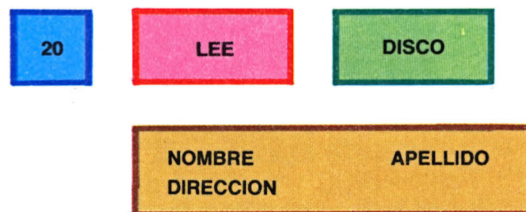
INSTRUCCION LEE

Funciones: permite la introducción de valores, numéricos o alfanuméricos, de los periféricos de entrada
- Video
- Cinta (C)
- Disco (D)

Sintaxis:



Ejemplo:



La unidad de entrada puede indicarse con sólo la primera letra, por lo que la instrucción puede convertirse en:

20 LEE V

Limitaciones: puede utilizarse un máximo de 10 campos

Diagnóstico: en fase de introducción se controlan los nombres de la unidad de salida (sólo la 1.ª letra) y la existencia en las definiciones de los campos utilizados

El valor a asignar al campo Número debe ser introducido por el operador, y de esta manera es posible anular, insertar o modificar cualquier instrucción de la misma manera que se gestionan los números de línea en el Basic. Durante la introducción aparecen máscaras sucesivas, cada una de ellas dedicada a una sola instrucción, hasta que el usuario introduce la palabra FIN, la cual termina la fase.

Todas las instrucciones introducidas pueden presentarse en pantalla con la voz 3 (Listado) del menú principal, obteniendo un listado de la forma representada en la última foto de la página 2357 y en la primera de la página 2358. Este programa realiza las siguientes funciones:

- Lectura del video, línea 10 LEE V (V = Video) de:
  - NOMBRE
  - CODIGO
  - CALLE
- Si CODIGO > = 2000, termina la ejecución (línea 15 DECIDE..., el último valor, 99, es la línea a la que se transfiere el control si la instrucción DECIDE tiene resultado positivo).
- Si no es así, escribe los datos en el disco, línea 20 ESCRIBE D (D = Disco)
- El conjunto se repite 15 veces, línea 30 REPITE 15, y los números que aparecen después de la instrucción (10, 15, 20) representan los procedimientos a llamar en el bucle. La palabra FIN, en este caso, representa el final del bucle y no del programa, que en cambio está en la línea 99.

En este ejemplo se hace evidente el ahorro de tiempo que se obtiene utilizando el programa en lugar de las instrucciones Basic. Además, el programa es «autodocumentado», puesto que todas las funciones que realiza se indican clara y sintéticamente. Si estuviese escrito en Basic, disponiendo solamente del listado, sería muy difícil identificar las funciones que realiza y, viceversa, en este lenguaje quedan muy claras.

Además, también es válido lo contrario, o sea si hay que construir una aplicación, en Basic es necesario pasar a los diagramas después de la fase de análisis. En cambio, en este caso, no se necesitan ulteriores desarrollos, puesto que el programa nace de la sencilla relación de las funciones a realizar.

INSTRUCCION ESCRIBE

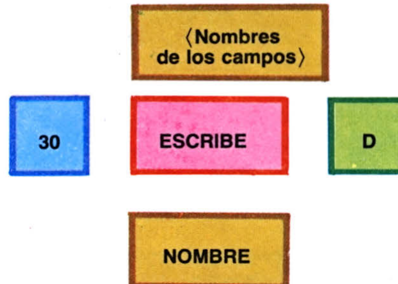
Funciones: genera la introducción en el periférico de salida especificado valores contenidos en los campos indicados. Los periféricos previstos son:

– Video (V)
– Cinta (C)
– Disco (D)

Sintaxis:



Ejemplo:



La instrucción genera la memorización en disco del contenido del campo NOMBRE en el fichero especificado en la fase de definición

Limitaciones: las mismas que las de la instrucción LEE
Diagnóstico: como LEE

Con respecto a esta instrucción debe observarse que no se ha previsto el número de registros en que escribir el dato. Este valor debe determinarse con otros procedimientos, como por ejemplo una búsqueda.
Además, como periférico de salida no está prevista la impresora. Esta carencia puede superarse fácilmente incluyendo la sigla I (Impresora) y una subrutina que gestione la impresión

INSTRUCCION BORRA

Funciones: permite el borrado de la pantalla o de un fichero

Sintaxis:



Ningún otro parámetro

Ejemplo:



Produce el borrado de la pantalla

Limitaciones: ninguna
Diagnóstico: sólo el control de la unidad

INSTRUCCION BUSCA

Funciones: permite realizar elecciones, mediante comparaciones, en los registros de un fichero (en disco o en cinta). Como la memoria de masa (cinta o disco) se define al principio, no es necesario que venga especificada en la instrucción, por lo que, con respecto a las anteriores, no necesita el campo Parámetro (este campo, si se introduce, no se tiene en cuenta)

Sintaxis:



Ejemplo:



En el fichero definido anteriormente busca el primer registro en el que el campo CIUDAD (también definido con anterioridad) contiene el valor "MADRID"

Limitaciones: los operadores que pueden utilizarse son >, <, =, < =, > =, < >
La selección puede realizarse en un solo campo cada vez

Diagnóstico: El nombre del campo ha de estar entre los definidos

N.B.: En la versión C-64, las dos comillas están sustituidas por una comilla

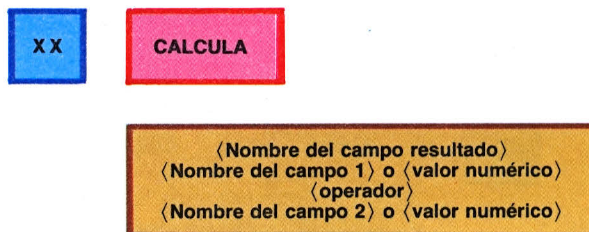
Otra ventaja del uso de este macrolenguaje es la notable estructuración que puede darse a los programas. Un software es «estructurado» cuando para cada función se utiliza una parte bien definida por sí misma. Esta implantación permite producir un software que puede modificarse fácilmente y adaptarse a distintas exigencias y, por tanto, es la actual tendencia seguida por todos los productores de software.
Entre los aspectos negativos del lenguaje Basic, también hay el de la dificultad de estructuración. Su lógica se basa principalmente en la ausencia de subrutinas como entidades en sí mismas, lo cual conduce muy a menudo a producir un software «desordenado», con muchas instrucciones del tipo GO TO..., que son lo menos estructurado que puede utilizarse.
Desde este punto de vista, son mucho mejores los lenguajes Fortran, Cobol y los más modernos, que se han creado con esta finalidad, como es el caso del Pascal. Si bien no constituye un lenguaje, el procedimiento descrito permite desarrollar aplicaciones de manera extremadamente estructurada, puesto que para cada función hay prevista una subrutina que se contempla como una «caja» a la que deben proporcionarse los valores (parámetros de entrada) para obtener otros en la salida.
En las figuras de las páginas 2345 y 2350 se han representado las informaciones principales acerca de las instrucciones previstas y algunos ejemplos de su utilización, los cuales son necesariamente muy esquemáticos y se complementarán e ilustrarán con detalle en los ejemplos de aplicación que siguen.

Empleo del software. Dada la particular estructura de este programa, en su empleo es necesario seguir una

INSTRUCCION CALCULA

Funciones: realiza el cálculo aritmético indicado. Obsérvese que no se ha introducido ninguna prioridad, por lo que el orden de ejecución es el de introducción

Sintaxis:



Ejemplo:



Suma en TOTAL el contenido de los campos IMPORTE y MOVIMIENTO

Limitaciones: los nombres de los campos deben ser todos definidos. Las operaciones previstas son: +, −, /, *

Diagnóstico: controla la existencia de las definiciones y de los operadores utilizados

determinada metodología. Ya se ha indicado la necesidad de realizar la fase de análisis con métodos y objetivos diferentes a los previstos para la escritura de programas en Basic.
Para ilustrar el método correcto de enfoque para afrontar un problema de programación con este software, es necesario tener bien presentes las finalidades y las limitaciones del procedimiento.
El primer objetivo es el didáctico. El procedimiento, o mejor, el conjunto de procedimientos reunidos en el programa, permite que cualquiera escriba inmediatamente los primeros y sencillos programas, porque sólo es necesario realizar correctamente el análisis. Este aspecto no debe ser despreciado de ninguna manera, porque es fundamental en la preparación cultural y, en general, en la formación. El estudio de un lenguaje de programación, incluso sencillo como el Basic, requiere empeño y continuidad, a veces tan fuertes, que pueden inducir a despreciar el otro aspecto fundamental del procedimiento: la lógica.
En la preparación de un programa, muy a menudo se salta la fase de análisis, e incluso a veces el trazado de los diagramas de flujo. De esta manera se obtiene un programa que, si bien realiza las funciones requeridas, no puede ser gestionado porque está privado de organización, o sea, no está estructurado. En cambio, siguiendo este método, se hace evidente la necesidad de tener bien claros los distintos pasos a realizar, y para cada uno se activa una adecuada rutina dedicada al mismo. En consecuencia, se tiene un elevado grado de estructuración y de modularidad.
Por tanto, esta metodología acostumbra al usuario a aislar los factores principales de una aplicación y a resolverlos lógicamente, por lo que constituye una buena introducción al estudio de la programación. Naturalmente, el software también puede utilizarse por

quien ya conoce los lenguajes de programación, puesto que sólo se trata de desarrollar algunas aplicaciones de manera sencilla e inmediata.
Finalmente, obsérvese que aislando las distintas subrutinas que realizan las funciones previstas, se tiene a disposición una «biblioteca» completa de software parametrizada y utilizable en cualquier aplicación.
Como ya se ha indicado, la mejor manera de desarrollar una aplicación con este software consiste en identificar las funciones elementales que la componen, y en activar los correspondientes procedimientos.
Por ejemplo, si se quiere construir una aplicación que permita gestionar un listín de direcciones, las funciones de base serán:

1 / Introducción datos, o sea formación del archivo
2 / Búsqueda, en base a cualquier campo, para la extracción de una determinada dirección (o sea que esta función es la interrogación del archivo)

Las dos funciones son completamente diferentes, y pueden realizarse en dos programas distintos.

Introducción. Para realizar esta función, que es un programa en sí misma, es necesario:

1 / Definir los campos de datos, o sea cómo estará constituido el archivo
2 / Definir el fichero, el nombre y la longitud
3 / Introducir los datos

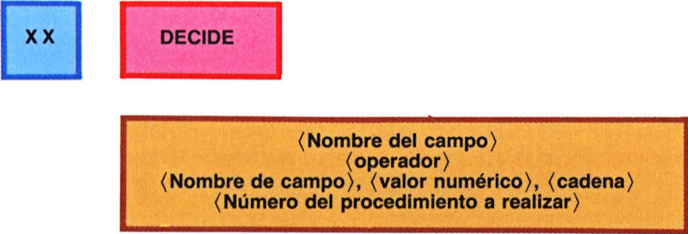
Las dos primeras funciones están contenidas en el procedimiento de definición, y la tercera requiere algunas instrucciones.
Para realizar las funciones de introducción de datos es necesario:

1 / Leer en el video: procedimiento LEE
2 / Comprobar si el operador desea terminar la introducción: procedimiento DECIDE (por ejemplo, introduciendo la palabra FIN)
3 / Escribir en el disco todo lo que se ha leído en la pantalla: ESCRIBE
4 / Repetir el bucle: LEE, DECIDE, ESCRIBE un núme-

INSTRUCCION DECIDE

Funciones: activa el procedimiento indicado al cumplirse la condición especificada

Sintaxis:



Ejemplo:



Envía a ejecución el procedimiento número 20 si el campo CIUDAD contiene el nombre MADRID

Limitaciones: los campos deben ser todos definidos. Los operadores previstos son >, <, =, >=, <=, < >. No realiza controles de congruencia, por lo que si el campo es definido, contiene caracteres y el valor de comparación es numérico, no se tiene diagnóstico

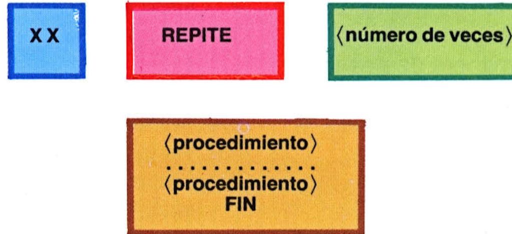
Diagnóstico: sólo en la definición de los campos

N.B.: en la versión C-64, las dos comillas están sustituidas por una comilla

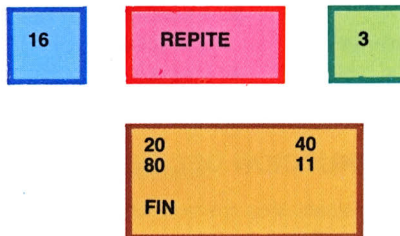
INSTRUCCION REPITE

Funciones: permite el desarrollo de un bucle para el número de veces indicado. En el bucle pueden reclamarse otros procedimientos hasta un máximo de 9

Sintaxis:



Ejemplo:



Los procedimientos 20, 40, 80 y 11 se repiten 3 veces. La palabra FIN es necesaria para indicar dónde termina el bucle y no puede omitirse

Limitaciones: los eventuales cálculos a realizar en el interior del bucle deben escribirse en un procedimiento y llamarse indicando su número

Diagnóstico: sólo en los nombres de los campos

Obsérvese que en esta instrucción, el campo (Parámetro), que normalmente contiene la indicación del periférico interesado, en cambio se utiliza para especificar el número de veces que debe repetirse el bucle

ro de veces igual al número de registros asignados al fichero

Terminado este análisis, mínimo, puede escribirse el programa simplemente completando las instrucciones (4 en total) con los oportunos parámetros.

Interrogación. Estructurando esta función como un programa en sí misma, es necesario introducir nuevamente las definiciones anteriores, por lo que las dos fases de introducción son idénticas.
En realidad, si la función se realiza de manera separada, puede evitarse la segunda introducción de las definiciones. Efectivamente, basta con escribir de nuevo las instrucciones con los mismos números de línea que las anteriores para obtener su sustitución, con lo que permanecen las definiciones y no deben introducirse.
El desarrollo del programa, o sea las instrucciones, en cambio, debe escribirse en base a la nueva utilización del archivo. La finalidad de este programa (recuérdese que su fisonomía de nuevo programa sólo es para sencillez, porque podría tratarse de una extensión del anterior) es la de buscar en un fichero la primera dirección correspondiente a una determinada ciudad, por lo que las funciones que se deben realizar son las siguientes:

- Introducción del nombre de la ciudad a seleccionar, procedimiento LEE (en la pantalla)
- Búsqueda del primer registro que contiene el valor introducido (en el campo correspondiente), procedimiento BUSCA
- Escritura en la pantalla de todos los datos contenidos en el registro así seleccionado, procedimiento ESCRIBE

De nuevo vuelve a llamarse la atención sobre la extremada sencillez y sinteticidad de las instrucciones. Por ejemplo, la función

20 BUSCA CIUDAD

para ser realizada en Basic, habría necesitado varias decenas de instrucciones (determinación del posicionado del campo en el registro, apertura del fichero, bucle de lectura, extracción del campo y comparación), además de presentar dificultades de interpretación. Sin embargo, téngase en cuenta que el lenguaje Basic no es superado por esta aplicación, pues todo el software es en Basic. Es decir, se produce la misma relación que hay entre el Basic y el Assembler: en Basic, algunos símbolos (nombres de las instrucciones, etc.) no hacen otra cosa que activar subrutinas escritas en Assembler. Análogamente, este «macrolenguaje» activa subrutinas escritas en Basic.

Módulo ilustrativo. La elasticidad del programa también permite su empleo para generar una aplicación terminada e ilustrar el propio programa. La estructura y el contenido de esta aplicación pueden ser definidos por el usuario. A continuación se presenta un ejemplo destinado solamente a proporcionar ulteriores explicaciones sobre el uso del programa. Ya se ha puntualizado que el mejor enfoque, utilizando este software, consiste en realizar el análisis «por funciones», o sea en definir paso a paso, con palabras, lo que se desea obtener sin entrar en detalles de «cómo obtenerlo», porque las macroinstrucciones permiten una traducción inmediata a nivel máquina.

En este ejemplo se ha tomado como objetivo la ilustración de las instrucciones simples que constituyen el lenguaje. Por tanto, es necesario crear un archivo que contenga las explicaciones de cada instrucción, con acceso mediante una clave que indique la finalidad de la instrucción. Es decir, la aplicación debe poder leer en la pantalla la función sobre la que se desean explicaciones, y presentar la correspondiente instrucción. Por tanto, para cada instrucción, el archivo debe contener:

— FINALIDAD
— SINTAXIS
— PARAMETROS
— NOTAS
— EJEMPLO

Por ejemplo, la instrucción LEE puede ilustrarse así:

FINALIDAD	:	Lectura de datos en pantalla
SINTAXIS	:	LEE VIDEO
PARAMETROS	:	Nombre de un campo
NOTAS	:	.....
EJEMPLO	:	LEE NOMBRE

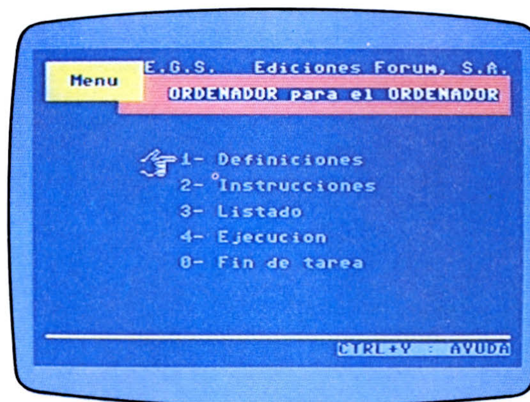
En la fase de utilización, al introducir por ejemplo la frase «Lectura datos pantalla» (a memorizar en el campo FINALIDAD), el programa realiza la búsqueda en el fichero y presenta las informaciones contenidas en el mismo. La aplicación está estructurada en dos fases. En la primera se introducen los datos que constituirán el archivo, y en la segunda es necesario buscar un determinado registro en base al contenido de un campo introducido por el usuario. Por tanto, se trata de una aplicación muy similar a la precedente (listín de direcciones), articulada en dos programas: el primero de creación, utilizando para los nombres de los campos los representados (FINALIDAD, SINTAXIS, etc.), y el segundo de búsqueda en el campo FINALIDAD, por lo que debe prever la introducción del dato a buscar (en el campo FINALIDAD) y la instrucción BUSCA.

El conjunto podrá incluirse en un bucle (instrucción REPITE) para la presentación sucesiva de una serie de datos, o puede agotarse con una sola búsqueda.

Obsérvese que después de haber creado el archivo, nada impide desarrollar aplicaciones utilizando campos diferentes de FINALIDAD, obteniendo así distintos métodos de acceso; por ejemplo, utilizando el campo SINTAXIS, se puede tener la comprobación del conocimiento del lenguaje realizando la comparación entre el dato introducido por el usuario y los datos memorizados en el disco.

Finalmente, obsérvese que esta misma estructura puede utilizarse para comprobar el conocimiento de cualquier otro lenguaje de programación.

Menú principal. La selección de la función deseada, como de costumbre, se obtiene introduciendo el número correspondiente.
La primera función a activar es la 1 (Definiciones), siguen la 2 (Instrucciones) y, al final, la 4, o sea el desarrollo del programa. La voz 3 (Listado) puede activarse en cualquier momento.



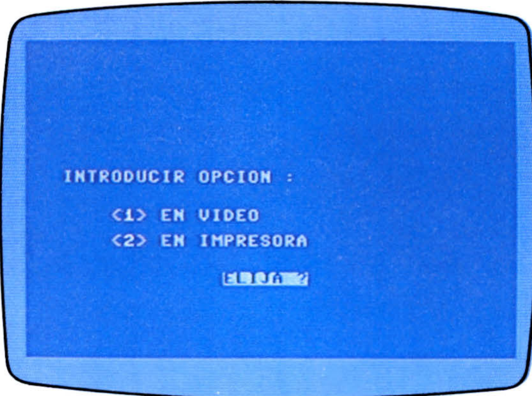
Definiciones. Se ha seleccionado la voz 1 y el sistema presenta la máscara para la introducción de los parámetros correspondientes a los campos que se utilizarán a continuación. En esta fase hay que definir tanto los campos a transferir al fichero como los necesarios sólo para procesos intermedios. La elección entre memorizar o no se realiza con la máscara siguiente.

Terminada la primera parte de la fase de Definiciones, hay que introducir los parámetros correspondientes a la gestión del fichero datos, como nombre y longitud (en número de registros). En la parte inferior de la máscara, el programa indica todos los nombres de los campos definidos anteriormente; introduciendo la letra "S", el usuario debe indicar los que desea memorizar.

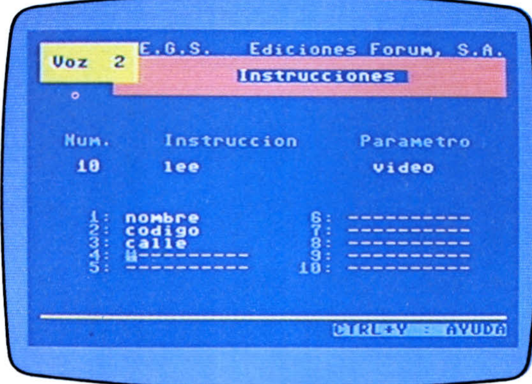
Al final de las Definiciones, o sea después de haber completado la máscara anterior, hay que indicar qué tipo de soporte se utilizará para el fichero. Respondiendo "S" a la pregunta presentada en la máscara de aquí al lado, se activa la gestión del disco.

Instrucciones. Todas las instrucciones se introducen con la misma máscara, naturalmente variando el formato según el tipo de instrucción.

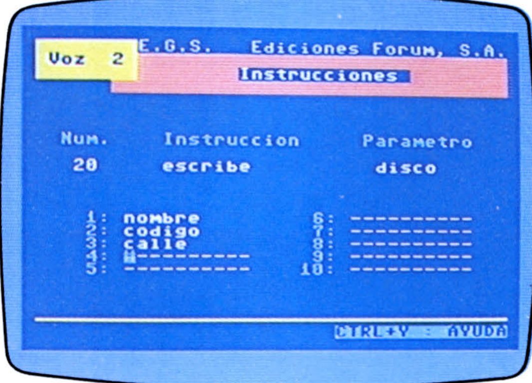
Listado. Seleccionando la voz 3 se tienen dos opciones: listado en pantalla o en impresora. La elección entre una y otra se obtiene respondiendo 1 o 2 a la pregunta presentada en la máscara.



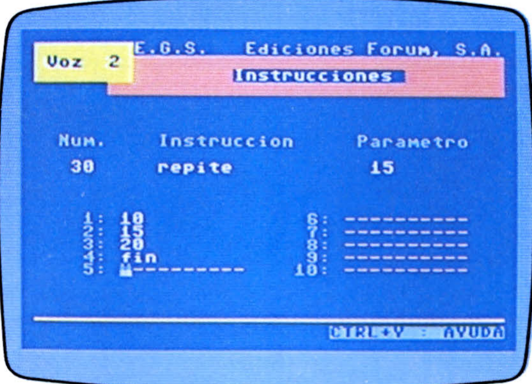
Sintaxis de la instrucción LEE. La foto muestra la estructura de la instrucción LEE direccionada a la pantalla. La línea es la número 10 y los campos, que serán leídos por el ordenador, son: nombre, código, calle. El tipo, la longitud y su posición en la pantalla deben haberse definido previamente (durante el desarrollo de la voz Definiciones).



Sintaxis de la instrucción ESCRIBE. Direccionada al disco, y por tanto destinada a memorizar en el fichero definido previamente los valores contenidos en los campos especificados, esta instrucción y la precedente constituyen un ejemplo de programa que puede leer en la pantalla algunos datos y memorizarlos en el disco. Obsérvese que la adquisición desde la pantalla (instrucción LEE anterior) contiene una complicada data entry. Para desarrollar en Basic un programa que proporcionara los mismos resultados, serían necesarias varias decenas de instrucciones.



Instrucción REPITE. Las dos fotos anteriores muestran las instrucciones necesarias para gestionar un grupo de datos, o sea una sola memorización en disco. Utilizando la instrucción REPITE, puede generarse un bucle que active las dos instrucciones anteriores para un número cualquiera de veces.
La instrucción REPITE mostrada en la foto genera un bucle que reclama 15 veces las instrucciones 10, 15 y 20. La palabra FIN que aparece en la máscara constituye el final del bucle, no del programa.



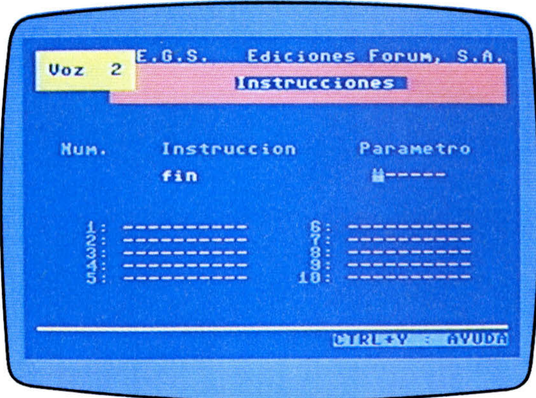
Instrucción DECIDE. El usuario desea "saltar" a la instrucción n.º 99 (que siempre indica el final del programa) al cumplirse la condición nombre = "fin". Este método puede utilizarse para interrumpir un bucle antes de cumplirse el número de repeticiones previstas.

Instrucción CALCULA. Permite realizar algunas operaciones sencillas. La foto muestra el cálculo de un valor (en el campo Total) como la suma de los contenidos de dos campos de nombres "uno" y "dos".

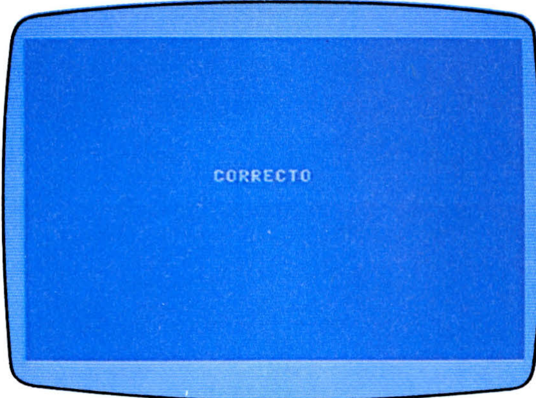
Instrucción BORRA. La instrucción se dirige al video y, por tanto, no contiene otros parámetros.

Instrucción BUSCA. El usuario desea seleccionar, entre todos los datos presentes en el fichero, el correspondiente al valor "Madrid" en el campo ciudad. Utilizando esta instrucción puede realizarse una selección de datos en base al contenido de cualquier campo. Es decir, se dispone de una estructura que tiene tantas "claves" de acceso como campos lo constituyen.

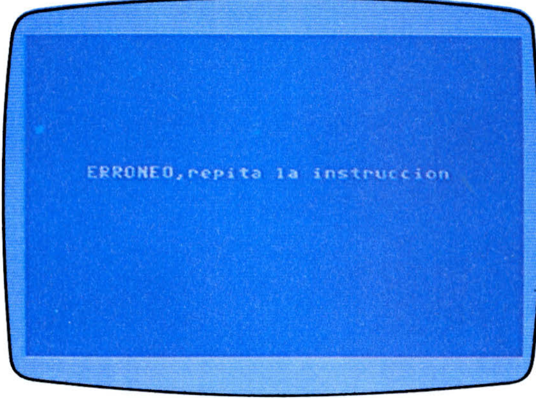
La foto muestra la instrucción que debe terminar cada programa, constituida por la palabra "fin" e insertada en la línea 99.



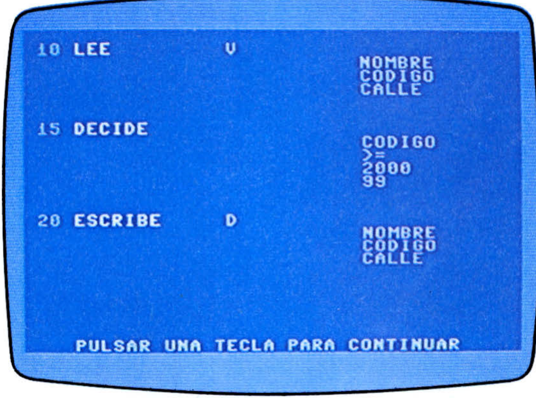
Al final de cada instrucción, el procedimiento realiza un control de sintaxis y emite un diagnóstico. La foto muestra el caso de una instrucción introducida correctamente.



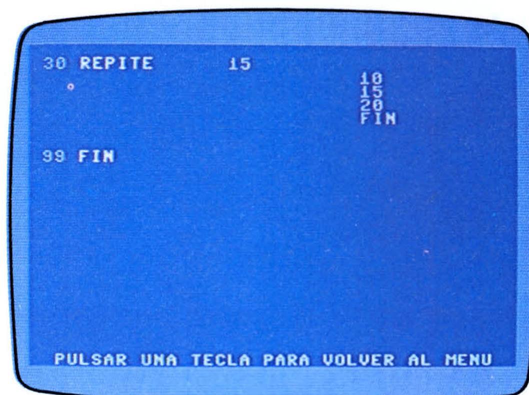
Si la sintaxis no es correcta o se emplea un nombre de campo no definido, la introducción es anulada y el usuario debe volver a escribir toda la instrucción.



La foto muestra una parte del listado en la pantalla. Para permitir su lectura, todo el listado se presenta en partes sucesivas con espera de un acuerdo (introducción de un carácter)



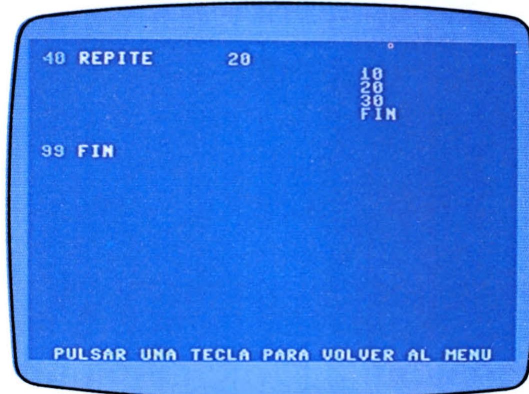
Después de la introducción, que activa el acuerdo para proseguir la presentación del listado, se muestra la parte siguiente del programa.



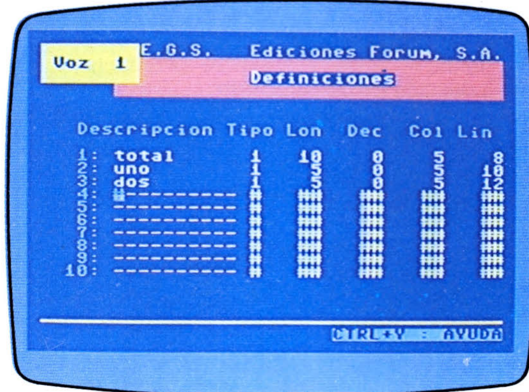
La foto muestra la primera parte de un programa de ejemplo para la gestión de un listín. La línea 10 permite la introducción, desde la pantalla, de los datos. La 20 se utiliza para terminar el programa antes de las 20 direcciones previstas (ver la foto siguiente) bajo la condición NOMBRE = "FIN".
La línea 30 permite la memorización en el disco de los datos introducidos (leídos por el ordenador) con la línea 10. Obsérvese que las unidades de I/O (Video y Disco) pueden indicarse únicamente con la primera letra (V y D).



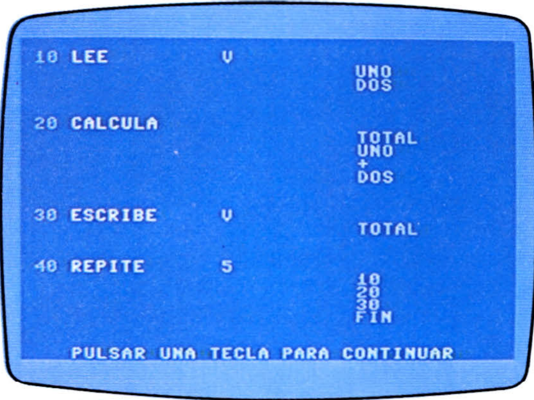
La foto muestra la segunda parte del programa, constituida por la instrucción REPITE, que activa un bucle sobre las instrucciones precedentes (10, 20, 30) para 20 veces, y por la última línea (99), que con el código FIN termina el programa.



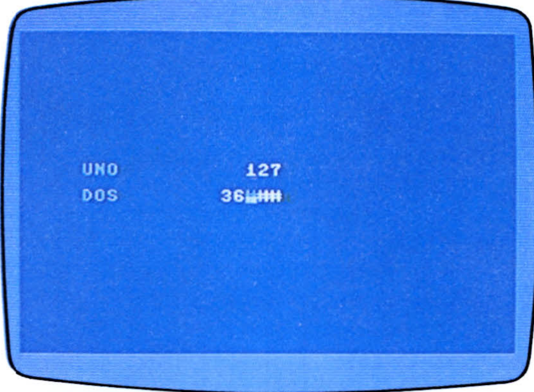
La foto muestra la fase de definiciones necesarias para generar tres campos numéricos de nombres TOTAL, UNO y DOS.



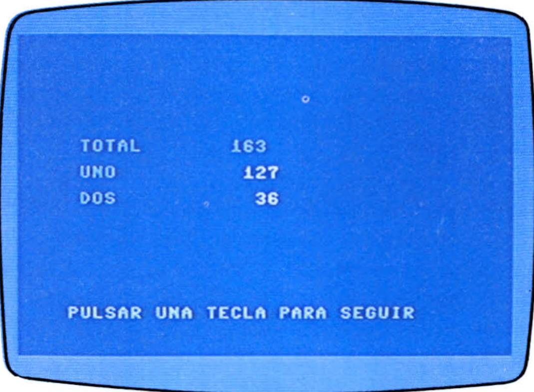
Los campos son gestionados en un programa que prevé la lectura en la pantalla de UNO y DOS (instrucción 10), el cálculo de TOTAL = UNO + DOS (línea 20), la presentación del resultado (TOTAL, línea 30), y la repetición de todo por 5 veces.



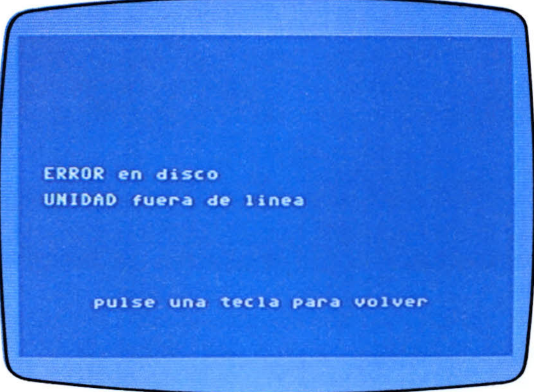
La foto muestra el resultado de la instrucción 10 de la máscara anterior. El usuario ha introducido el valor 127 en el campo UNO y está terminando de introducir otro valor en el campo DOS.



Al final de la introducción se calcula el total (línea 20) y se presenta el resultado (línea 30).



En el procedimiento hay incluidos algunos controles. La foto muestra el diagnóstico que aparece por errores en el disco.



La lógica del programa

La lógica del procedimiento puede resumirse en los siguientes pasos:

1 / El sistema debe poseer un conjunto de rutinas completamente parametrizadas, que, por área de datos, utilice las mismas variables para obtener fácilmente los intercambios de una a otra
2 / El usuario, mediante frases simbólicas (instrucciones), proporciona a la máquina la dirección de la rutina que desea activar. Los eventuales parámetros necesarios para su desarrollo se introducen en el área de datos comunes
3 / En este punto sólo hace falta un intérprete que analice las palabras, o frases, de direccionado y active la oportuna rutina

En la página siguiente se ha representado el esquema de principio del programa. La primera parte se dedica a las definiciones (campo y ficheros) y a la introducción de las instrucciones, y la segunda a su desarrollo. Durante la fase de definición y de introducción de las instrucciones se realiza un diagnóstico para evitar el uso de campos no definidos o de instrucciones no correctas, por lo que en la fase de ejecución no son necesarios otros controles. En realidad, esta metodología no impide la producción de errores durante la ejecución. El diagnóstico inicial puede detectar errores formales, precisamente del tipo de campos no definidos o de instrucciones no reconocidas, pero no puede determinar eventuales errores «lógicos», como por ejemplo la división por cero. Por tanto, este tipo de error es interceptado por el sistema operativo de la máquina (recuérdese que si bien el sistema utiliza un lenguaje particular, en realidad trabaja bajo Basic) y da lugar a los usuales diagnósticos que, además, forman el bloque de la ejecución. Sin embargo, como implantación puede incluirse una rutina de «trampa» para los errores que detecte y emita un diagnóstico preparado por el usuario en lugar de hacerlo el sistema. Esta implantación está estrechamente ligada al tipo de máquina utilizado, por lo que, por motivos de generalización, no se comenta en los listados presentados. La segunda fase, ejecución, puede resumirse en un bucle que realiza las siguientes funciones:

1 / Toma una instrucción de la memoria
2 / La interpreta (toma del nombre simbólico el puntero a la subrutina que realiza la función indicada)
3 / Llama la subrutina «de sistema» así identificada

Más adelante, en el gráfico de la página 2360, se ha representado un esquema que muestra cómo se utilizan los datos introducidos en la fase de definición (de los campos):

- **Nombre del campo** (p.e. DIRECCION): se utiliza para dos finalidades, presentar en las máscaras video la descripción del campo al que se refiere y como referencia en todas las instrucciones. Una vez definido y memorizado un nombre, todas las instrucciones y también las definiciones del fichero pueden utilizarlo. Por ejemplo, si en la instrucción LEE hay el nombre del campo DIRECCION, a través de los atributos memorizados conoce su longitud, su tipo y su posicionado en la pantalla y, por tanto, puede activar la rutina para su lectura (estructuralmente muy similar a la de la data entry)
- **Tipo** (p.e. 3): es un código numérico que indica el tipo al que pertenece el campo, con lógica y funciones análogas a las de la data entry
- **Longitud** (p.e. 20): define la longitud del campo. Se utiliza para la formación del registro (la longitud es igual a la suma de las longitudes de los campos que lo constituyen), para la extracción del valor del campo de un registro (por ejemplo, en el procedimiento BUSCA) y en la preparación y la gestión de la máscara video
- **Número de decimales** (p.e. 0): define las cifras decimales de los campos numéricos. Se utiliza en la gestión y la presentación de la máscara video. El formato exacto de escritura en el fichero se obtiene automáticamente de estas gestiones que insertan en la posición justa los símbolos (coma o punto) de separación entre la parte entera y la decimal del valor numérico
- **Columna y línea** (p.e. 2, 4): definen el posicionado del campo en la pantalla. Por tanto, sólo se utilizan en las funciones de I/0 correspondientes a este periférico

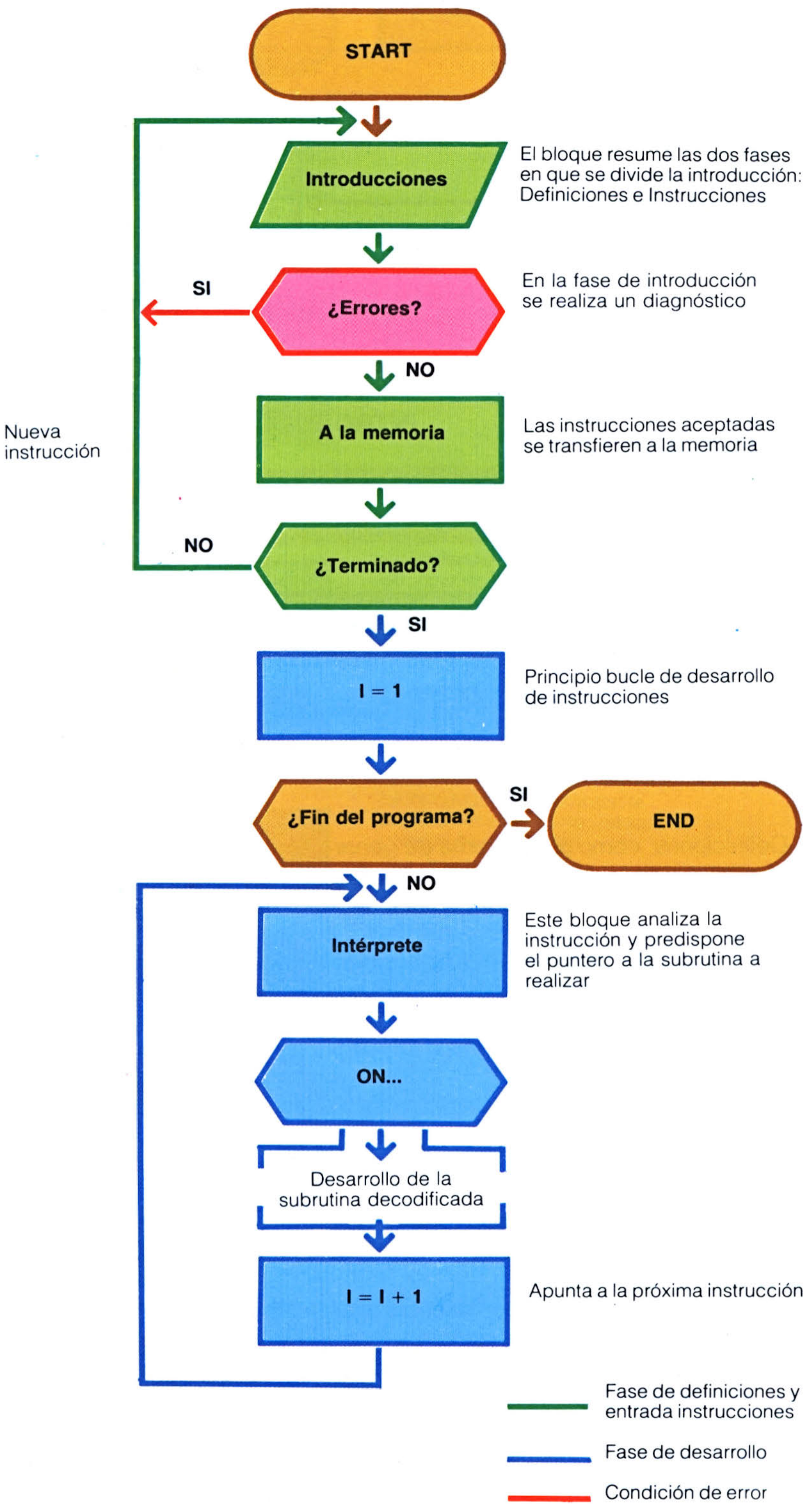
En lo que respecta al formato de las instrucciones, el significado de los distintos parámetros es:

- **Número** (p.e. 21): define la posición de la instrucción en el interior del área de memoria dedicada a contener el listado del programa. Su empleo es necesario para poder insertar líneas en un listado ya existente y, además, puede utilizarse como puntero en la decodificación y en la memorización del programa en el disco
- **Instrucción** (p.e. LEE VIDEO): mediante el nombre simbólico, identifica la subrutina a activar. El mecanismo de decodificación y de activación de la subrutina se ilustra más adelante
- **Nombres de los campos** (p.e. DIRECCION): determinan en qué campos, entre los definidos anteriormente, deben realizarse las diversas funciones especificadas en la subrutina identificada por la instrucción.

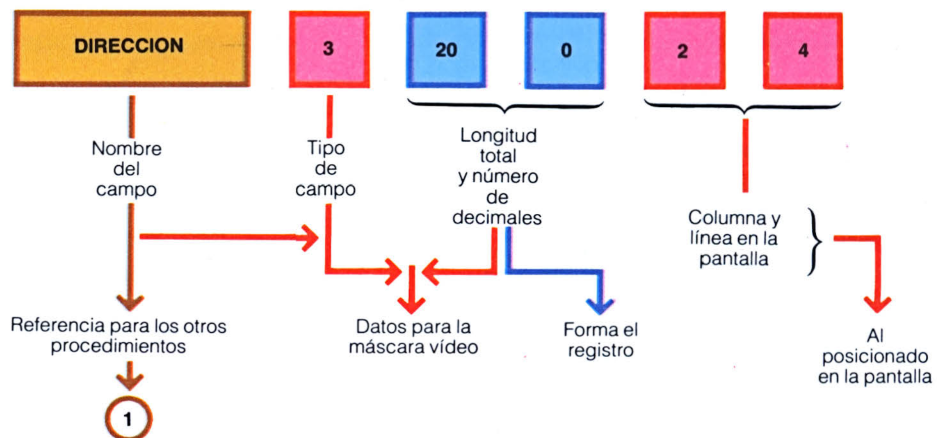
Fase de definición. Como ya se ha indicado, la primera función a activar durante el uso del software es el procedimiento de definición, a su vez dividido en dos fases. En la primera se definen los nombres simbólicos de todas las variables que se van a utilizar y de sus atributos. En la segunda se especifican cuáles, entre los introducidos, deben transferirse al disco. Esta segunda fase es necesaria para permitir el uso de variables «locales», o sea que puedan utilizarse para una aplicación particular sin interesar, directamente, los datos memorizados en el fichero.
Por ejemplo, se desea gestionar unas existencias de almacén con los siguientes campos:

- Nombre del artículo
- Existencia
- Costo

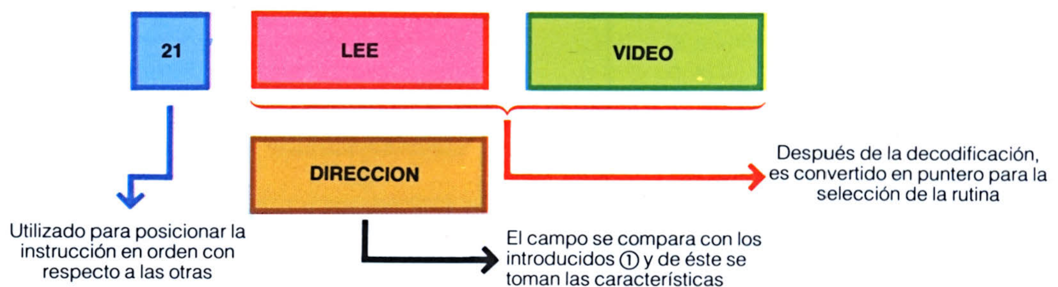
Los valores de los 3 campos previstos deben introducirse en la fase de generación para todos los artículos que se desea gestionar. Por tanto, la formación del archivo consiste en un bucle de lectura de vídeo y escritura en disco. Terminada esta fase, debe actualizarse la existencia, sumando la cantidad de llegada y restando la de salida. Por tanto, es necesario un nuevo campo (Movimiento) en el que escribir la cantidad a mover. Sin embargo, este valor no debe memorizarse, puesto que su uso se limita al desarrollo del cálculo de una nueva existencia. Por tanto es necesario un programa (diferente al de generación) en el que, además de los campos que componen el registro, se defina el campo de movimientos, cuyo contenido se utilizará para calcular la nueva existencia, que deberá memorizarse en el fichero. Este programa, por ejemplo para la llegada de nueva mercancía, se compone de los procedimientos:



DEFINICIONES DE LOS CAMPOS



INTRODUCCION INSTRUCCIONES

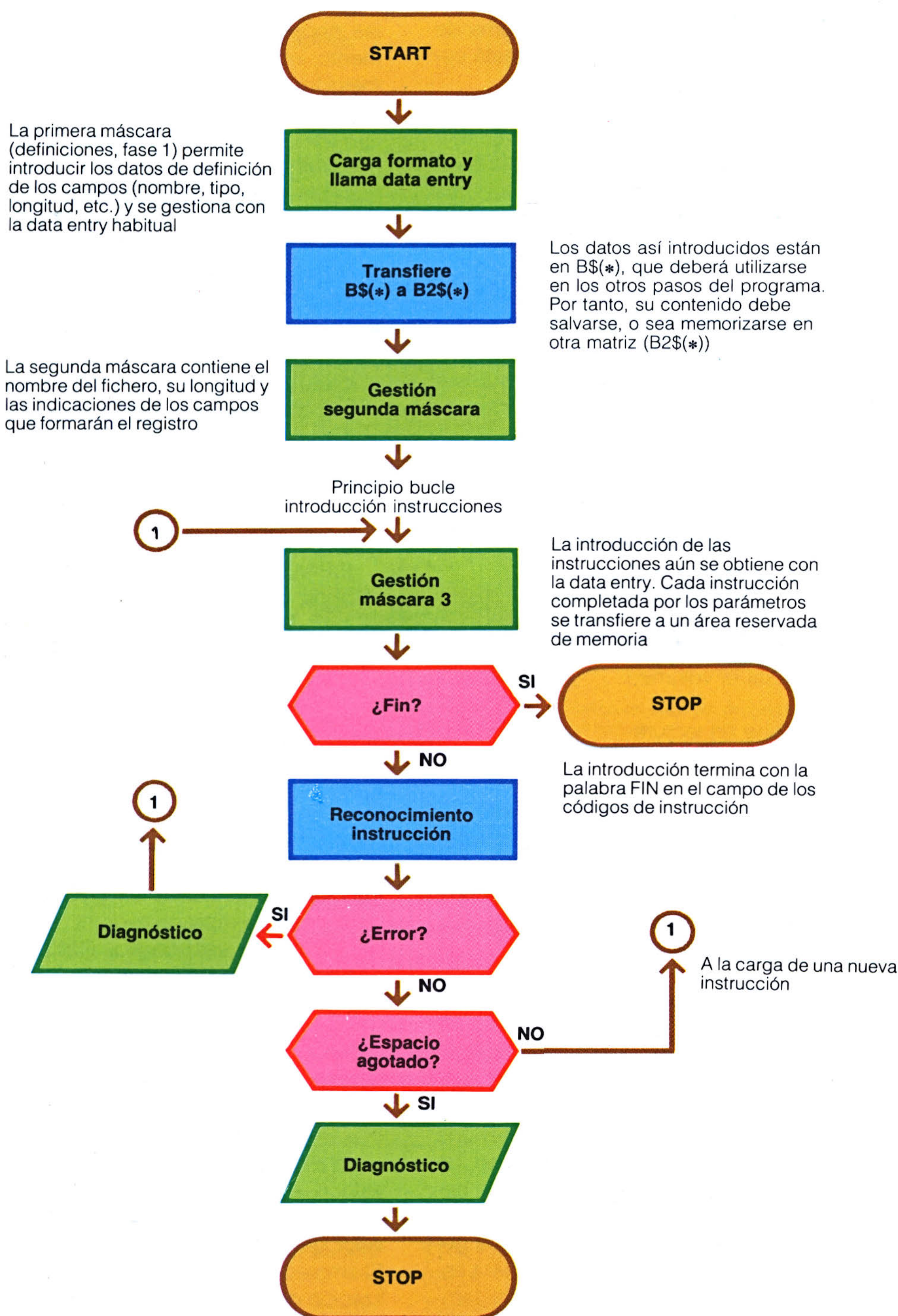


- Definiciones: como el programa de generación, pero con el campo MOVIMIENTO además, en el que se introducirá la cantidad a sumar o a restar. Este campo no deberá transferirse al fichero, por lo que en la segunda parte del procedimiento de definición, al lado del nombre MOVIMIENTO, deberá escribirse una N (no, no memorizar)
- Lectura del vídeo de los campos NOMBRE (del artículo) y MOVIMIENTO
- Búsqueda, en el fichero, del artículo (BUSCA NOMBRE)
- Cálculo de la nueva existencia (existencia leída en el disco + MOVIMIENTO)
- Memorización del nuevo dato, o sea de los campos definidos a transferir al disco, menos MOVIMIENTO.

Este ejemplo utiliza dos procedimientos similares para los dos tipos de movimientos previstos (entrada y salida), mientras que, definiendo un segundo campo «local», es posible incluir los dos cálculos en el mismo procedimiento. Por ejemplo, creando un campo en el que esté contenido el tipo de movimiento (entrada, salida), con el procedimiento DECIDE se puede determinar si el cálculo es una suma o una resta, unificando así los dos modos de desarrollo.

El diagrama de principio correspondiente al procedimiento de definición y de introducción de las instrucciones se ha representado en la página siguiente. Antes de proceder a su descripción, son necesarias algunas precisiones de carácter general. El software se ha desarrollado de manera que gestione tanto la unidad de disco como la de cinta: la particular estructura de este segundo periférico necesita siempre la carga en memoria de todo el fichero datos, por lo que para ello es necesario reservar una importante área de memoria, limitando además el número máximo de registros para cada fichero. En cambio, si sólo se tomase en consideración la unidad de disco, todos estos problemas quedarían superados. Además, y siempre por estos motivos, no se ha previsto la posibilidad de memorizar los programas para volverlos a utilizar, puesto que se trata de una función accesoria poco importante, ya que los programas escritos con este software se reducen a unas pocas líneas y, por tanto, es muy sencillo volverlos a escribir.
En el mismo gráfico se presentan los tres pasos del procedimiento. Inicialmente se activa la definición de los nombres y de los atributos de los campos que se prevé van a utilizarse. Esta parte, como las otras, sólo es una aplicación particular de la subrutina de gestión

DEFINICIONES E INTRODUCCION INSTRUCCIONES: DIAGRAMA DE PRINCIPIO



En la salida, los principales parámetros son:
B2$(*) Contiene los nombres de los campos, el tipo, la longitud y todos los demás atributos
P7$(*) Contiene los nombres de los campos utilizados en las instrucciones
SK$(*) Indica los campos que deberán transferirse al fichero

de la data entry, por lo que no son necesarios ulteriores comentarios. Sólo deben tenerse en cuenta eventuales definiciones preexistentes. Por ejemplo, en la gestión de almacén ilustrada anteriormente, en la definición correspondiente a la escritura del programa de actualización, hay que tener en cuenta lo que está escrito en el programa de generación.
En la actualización se necesita tan sólo el nombre del artículo para su identificación, la existencia y la cantidad movida; por tanto, con respecto al programa de generación, a fines del desarrollo del programa no es necesario definir el campo «Costo». Sin embargo, en realidad, este campo debe preverse aunque no se utilice, porque si no se hiciese así, este campo tendría una longitud que no correspondería al formato real del fichero, con los consiguientes errores.
Sin embargo, hay que tener en cuenta que esta posibilidad de redefinición puede aprovecharse para aumentar las potencialidades del programa, introduciendo una forma de estructuración de los datos.
Por ejemplo, supongamos que se quiere construir un archivo en el que memorizar los datos correspondientes a una biblioteca, y precisamente:

1 / Título
2 / Autor
3 / Materia tratada
4 / Tema
5 / Armario en que está el volumen
6 / Estante
7 / Persona que lo tiene en lectura
8 / Fecha en que se ha prestado
9 / Fecha de devolución

Algunos de estos campos presentan afinidad, por ejemplo, la materia tratada y el tema (campos 3 y 4) pueden resumirse como «claves de acceso», los campos 5 y 6 pueden convertirse en «posición», etc. Por tanto, el formato puede convertirse en:

1 / Registro (título + autor)
2 / Clave (materia + tema)
3 / Posición (armario + estante)
4 / Préstamo (suma de los campos 7, 8 y 9)

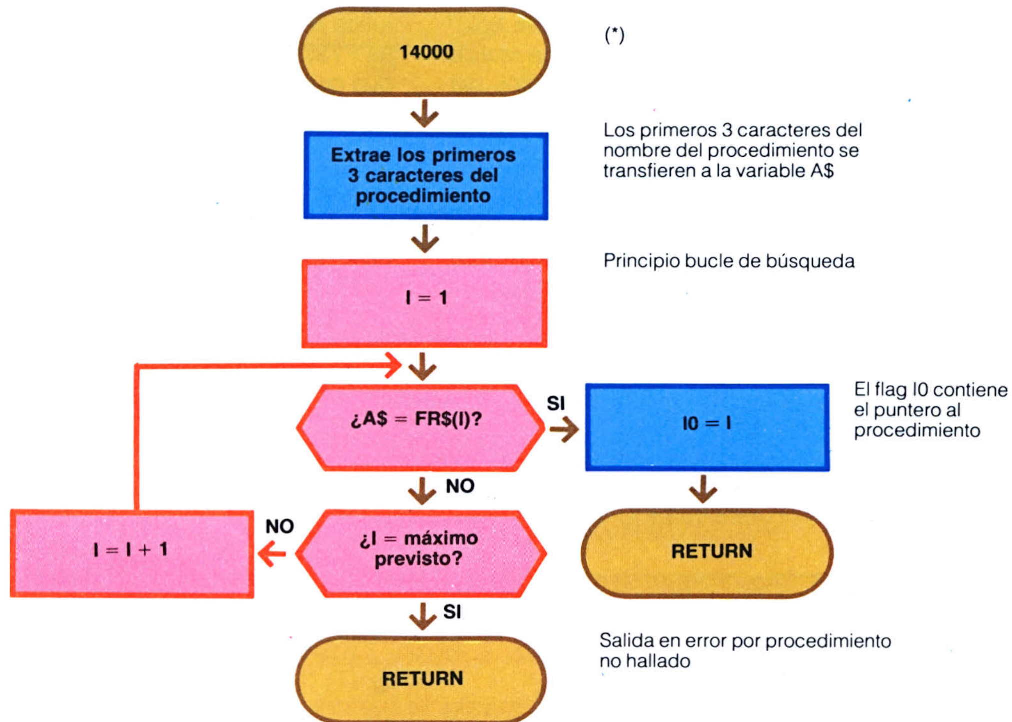
Naturalmente, los distintos campos deberán tener una longitud igual a la suma de los que los constituyen. La estructura de datos podrá redefinirse entonces en base a la aplicación particular.
Por ejemplo, si se desea buscar los libros escritos por un determinado autor, conviene adoptar la primera estructura activando el procedimiento BUSCA en el campo 2 (Autor), mientras que si se desea saber dónde se encuentra un determinado volumen, conviene la segunda, activando el procedimiento BUSCA en el campo 1 (Registro = título + autor).
Esta metodología es muy importante, puesto que el procedimiento BUSCA sólo puede utilizar un campo, y sin una oportuna estructuración de los datos sería demasiado limitativa y no permitiría más selecciones simultáneas. En cambio, con el método expuesto pueden redefinirse varias claves de acceso en función de las necesidades específicas de la aplicación. Finalmente, obsérvese que aplicando correctamente esta metodología, y con oportunas implantaciones en el software, es posible construir una Base de Datos de tipo «relacional», muy difícil de gestionar en Basic, costosa si se realiza con otros métodos, de difícil ejecución en sistemas pequeños y, por esto, prácticamente desconocida en los ordenadores domésticos.
En lo referente a la simbología adoptada en los diagramas de estas páginas, hay que recordar que:

- Los números de línea en que empiezan las distintas subrutinas se refieren únicamente a la versión C-64; para el MSX y el Apple deben considerarse meramente indicativos. Las subrutinas representadas, si bien son comunes a todos desde el punto de vista lógico, presentan notables diferencias de desarrollo; baste pensar por ejemplo en la generación de una cadena en blanco, que en algunas máquinas (sistema MSX) se resuelve con una sola instrucción (SPACE$(*)), mientras que en otras se necesita un bucle. En la gestión del disco también hay importantes diferencias
- No se han dado detalles de presentación ni de gestión de máscaras porque, después de su reiterado empleo en programas anteriores, se supone que están suficientemente claras.

Reconocimiento instrucción. En la segunda parte de las definiciones, en el momento de la introducción, cada instrucción es «reconocida», o sea se realiza un control para comprobar que la función requerida (instrucción) pertenece a las previstas. En la página siguiente se ha representado el diagrama de la subrutina que realiza este control. Cada instrucción es reconocida analizando únicamente los 3 primeros caracteres, después la subrutina la extrae y efectúa una comparación con el contenido de la matriz FR$(*), que contiene las siglas de las instrucciones previstas.
Si el control es positivo, la instrucción se da como válida y el flag I0 asume el valor numérico correspondiente a la posición en la que se ha hallado la instrucción en la matriz FR$(*); si no es así, I0 se mantiene en el valor 0. Por tanto, como respuesta de la rutina se tendrá el puntero I0 al valor 0 si se está en error o a un valor distinto si la instrucción es reconocida.
El empleo de este indicador no se limita sólo a la señalización de error, sino que puede extenderse a una codificación numérica de la instrucción.
Una primera y obvia implantación del programa consiste en incluir en él la función de memorización y relectura para su ulterior utilización. En esta fase, para ahorrar espacio, en lugar de memorizar las instrucciones en forma descriptiva, tal y como se introducen,

RECONOCIMIENTO DE INSTRUCCIONES

En la llamada, la matriz FR$(∗) debe contener las primeras 3 letras de los nombres de procedimientos activos



Los procedimientos se reconocen utilizando los tres primeros caracteres del nombre, por lo que las cadenas que constituyen la matriz FR$(∗) de comparación son todas de 3 caracteres

Nombre del procedimiento	FR$(∗)	Puntero (I0)
LEE	LEE	1
ESCRIBE	ESC	2
BORRA	BOR	3
BUSCA	BUS	4
CALCULA	CAL	5
DECIDE	DEC	6
REPITE	REP	7

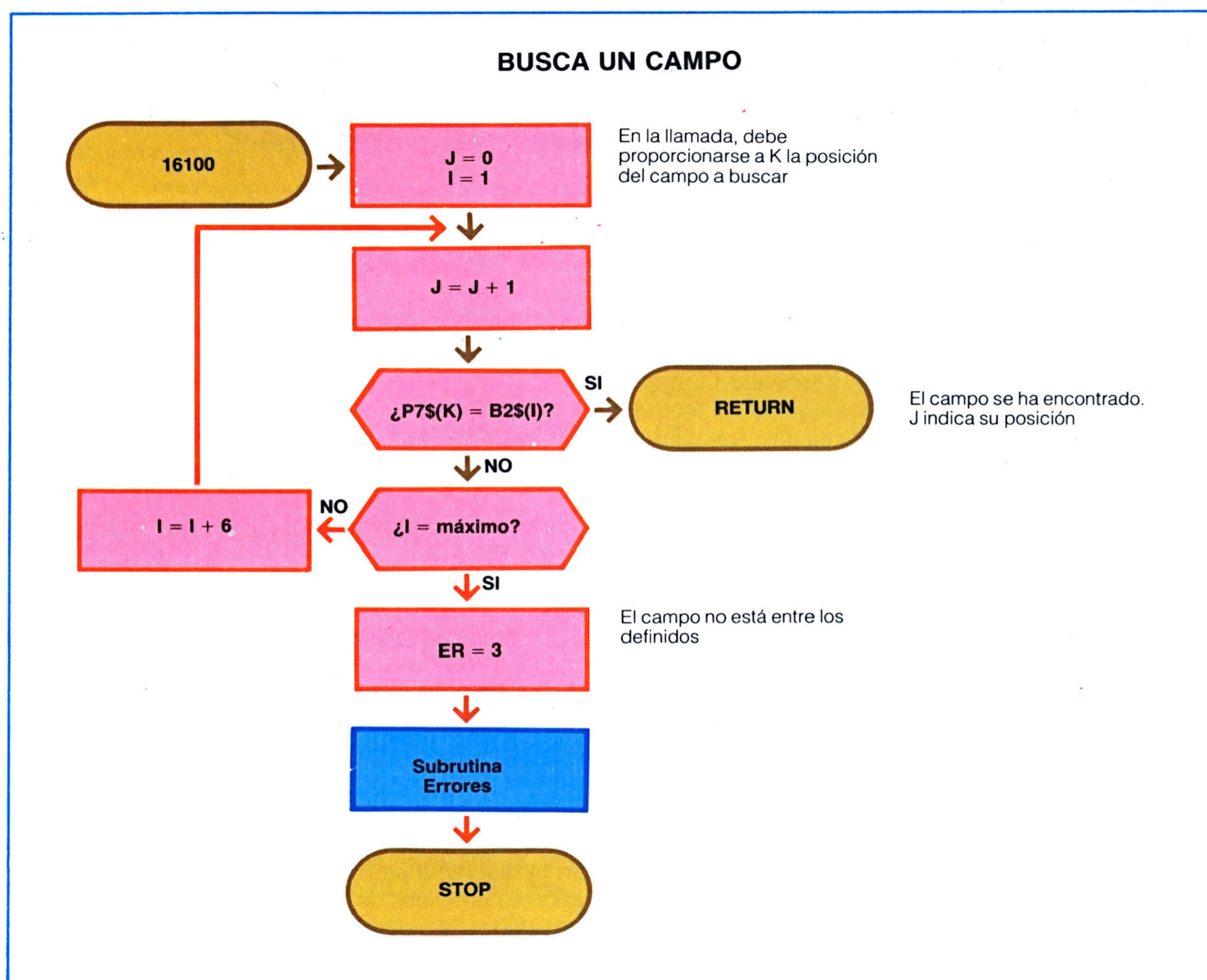
(*) En éste, así como en los siguientes diagramas, los números de línea se refieren únicamente a la versión C-64. Para el MSX y el Apple, deben considerarse puramente indicativos

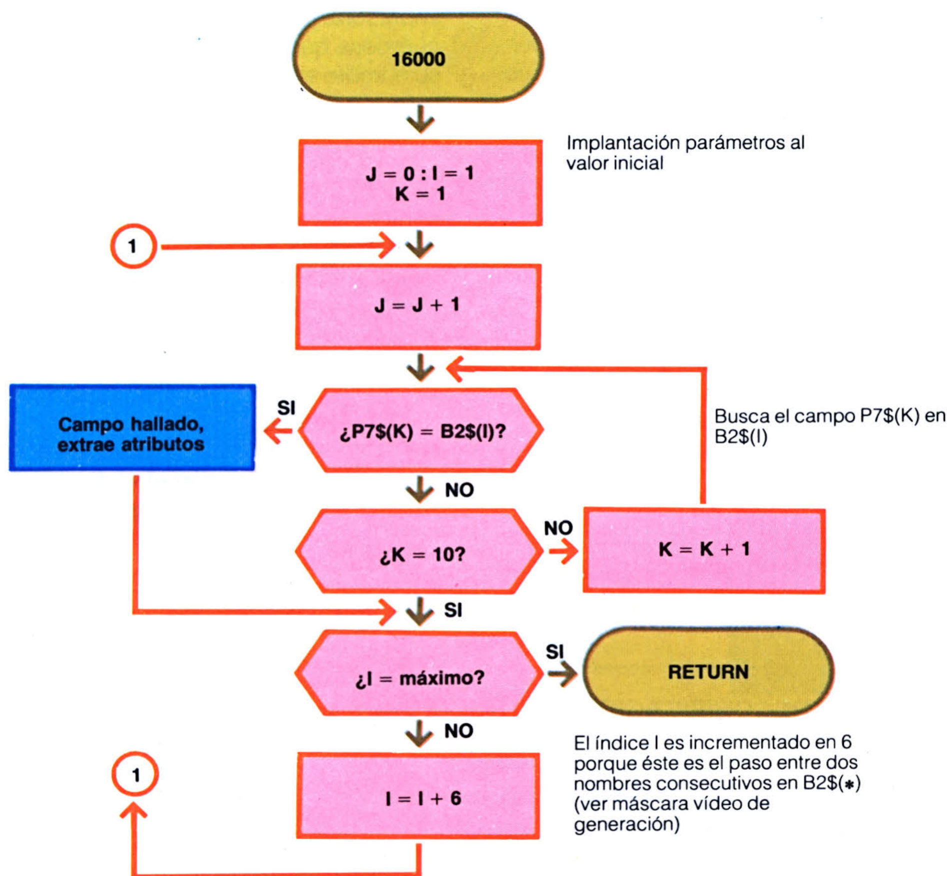
conviene transformarlas en numéricas y memorizar su valor. Por ejemplo, la instrucción LEE se convierte en 1, la BUSCA en 4, y así sucesivamente, con un ahorro de muchos bytes.
Sin embargo, al activar esta implantación, el listado que se obtiene después de la relectura ya no puede ser en «lenguaje llano», porque las instrucciones han sido sustituidas por un código numérico. Por tanto, para permitir su lectura es necesario regenerar el nombre simbólico, lo cual puede conseguirse, limitándose a los 3 primeros caracteres, utilizando la matriz FR$(∗).

Así, con el ejemplo anterior, para obtener las siglas de las instrucciones codificadas, basta con utilizar FR$(1), que contiene "LEE" y FR$(4), que contiene "BUS". Si no se desea utilizar esta forma abreviada, habrá que memorizar en FR$(∗) todo el nombre de la instrucción, obteniendo una reconstrucción completa.
En el diagrama se han representado las siglas y los valores numéricos de las 7 instrucciones previstas. En los listados, la matriz FR$(∗) tiene un dimensionado superior en previsión de nuevas instrucciones que podrían ser creadas e introducidas por el usuario.

Por último, obsérvese que en esta subrutina no se realiza ningún control sobre los campos y, por tanto, para asegurar la exactitud de la instrucción, el control debe hacerse en otro punto.

Búsqueda de un campo y extracción atributos. En la descripción de la subrutina anterior, se ha indicado la necesidad de prever un control de existencia de las definiciones de los campos utilizados en las instrucciones. Abajo se ha representado la subrutina que realiza esta función, y que se utiliza en muchos puntos del programa. En algunas instrucciones, el simple reconocimiento no es suficiente, y es necesario extraer también los atributos del campo utilizado. A este fin se ha creado la subrutina que se muestra en la página siguiente, que también incluye la función realizada por la anterior (comprobación de existencia). La subrutina de búsqueda de un campo pide, en la llamada, la implantación de la variable K al valor correspondiente al campo que se busca. Al escribir las instrucciones, las distintas líneas de la máscara se transfieren inicialmente al buffer de la pantalla B$(*), como si se tratase de una data entry normal. A la salida de la introducción, el buffer se desempaqueta en sus componentes, entre ellos, los parámetros asociados a la instrucción que son transferidos a la matriz P7$(*); en consecuencia, ésta contiene los nombres utilizados en la instrucción. En la llamada a la subrutina representada abajo, asignando un valor a la variable K se identifica el correspondiente nombre: por ejemplo, poniendo K = 3, la rutina realiza el control sobre el contenido de la cadena P7$(3). La comprobación se obtiene con un bucle que compara P7$(K) con todas las cadenas de la matriz B2$(*). Además, en la subrutina se utilizan los índices J e I. El primero se incrementa con cada paso en el bucle e indica, si la búsqueda ha tenido un resultado positivo, la posición del campo buscado en B2$(*). El segundo (I) es el índice del bucle que debe tener paso 6 (por este motivo se emplean dos índices), porque en la matriz B2$(*), los nombres de los campos distan 6 posiciones entre sí (cada nombre va seguido del tipo, la longitud, el número de decimales, la línea y la columna, o sea 5 parámetros) y para identificar el segundo nombre hay que avanzar con paso 6. La sub-





rutina representada arriba se desarrolla de manera análoga, con la diferencia de que el puntero K al campo a buscar es gestionado internamente. Efectivamente, la subrutina anterior determina la posición de un solo campo, por lo que necesita su indicación. La segunda extrae los parámetros de todos los campos cuyos nombres están contenidos en la matriz P7$(*), por lo que el valor de K debe variar entre 1 y el número máximo previsto, para poder así extraer todos los nombres. A medida que los campos son identificados, de ellos se extraen los atributos y se transfieren a la matriz B$(*), a la TP(*), etc.: así se obtienen los valores necesarios para el funcionamiento de la data entry o para la transferencia al disco.
Obsérvese que en estas, así como en las demás subrutinas previstas, las matrices B$(*), TP(*), etc., son consideradas áreas de trabajo, por lo que para realizar cualquier proceso hay que transferirles antes los datos a los que se refiere el proceso y después llamar la subrutina que lo ejecuta. De manera análoga, los eventuales resultados de cálculo se transfieren a una cadena de la matriz B$(*). Esta lógica a veces obliga a «salvar» los contenidos de B$(*), etcétera, que podrían ser «ensuciados» con procesos intermedios.

Decodificación y desarrollo de las instrucciones
En los diagramas que siguen se ilustran los principales aspectos de las subrutinas que interpretan y ejecutan las distintas instrucciones. También para éstas valen las advertencias anteriores, o sea:

- La gestión de las máscaras se indica sintéticamente con un solo bloque funcional, suponiendo que el lector conozca suficientemente las correspondientes subrutinas que, sin embargo, pueden ser analizadas en los programas anteriores.
- Muchas funciones se esquematizan en un bloque descriptivo único; para ver su detalle, diferente en cada máquina, hay que consultar los respectivos listados.

- En general, los nombres de las variables permanecen invariables en las distintas versiones. Sin embargo, no es una regla fija, porque a veces son necesarias algunas modificaciones debidas a las diferencias entre las distintas versiones del Basic.

Instrucciones LEE y ESCRIBE. Son las instrucciones de I/O previstas en este lenguaje. En la versión presentada no se incluye la gestión de la impresora, que puede ser implantada por el propio usuario siguiendo el trazado de las funciones realizadas para el vídeo.
En el gráfico de abajo se ha representado la subrutina de las dos instrucciones. La diferencia entre la una y la otra (lectura/escritura) viene indicada por el valor de I0, mientras que los otros parámetros no varían.
En ambos casos, las primeras funciones son la búsqueda de los campos reclamados en la instrucción y la extracción de sus atributos: esto se realiza utilizando la subrutina de la página anterior, que, en la salida, restituye las matrices B$(*), TP(*), etc., con los valores correspondientes a los campos que se van a utilizar.
Sigue la selección de la oportuna rutina en base al periférico seleccionado entre:

1 / Vídeo: se activa el bloque de gestión de las máscaras (data entry) con la implantación de la matriz TP(*), que se pone al valor correspondiente a la presentación si I0 vale 2

2 / Cinta o disco: activan las dos subrutinas correspondientes. La distinción entre lectura y escritura está en el interior de las propias subrutinas.

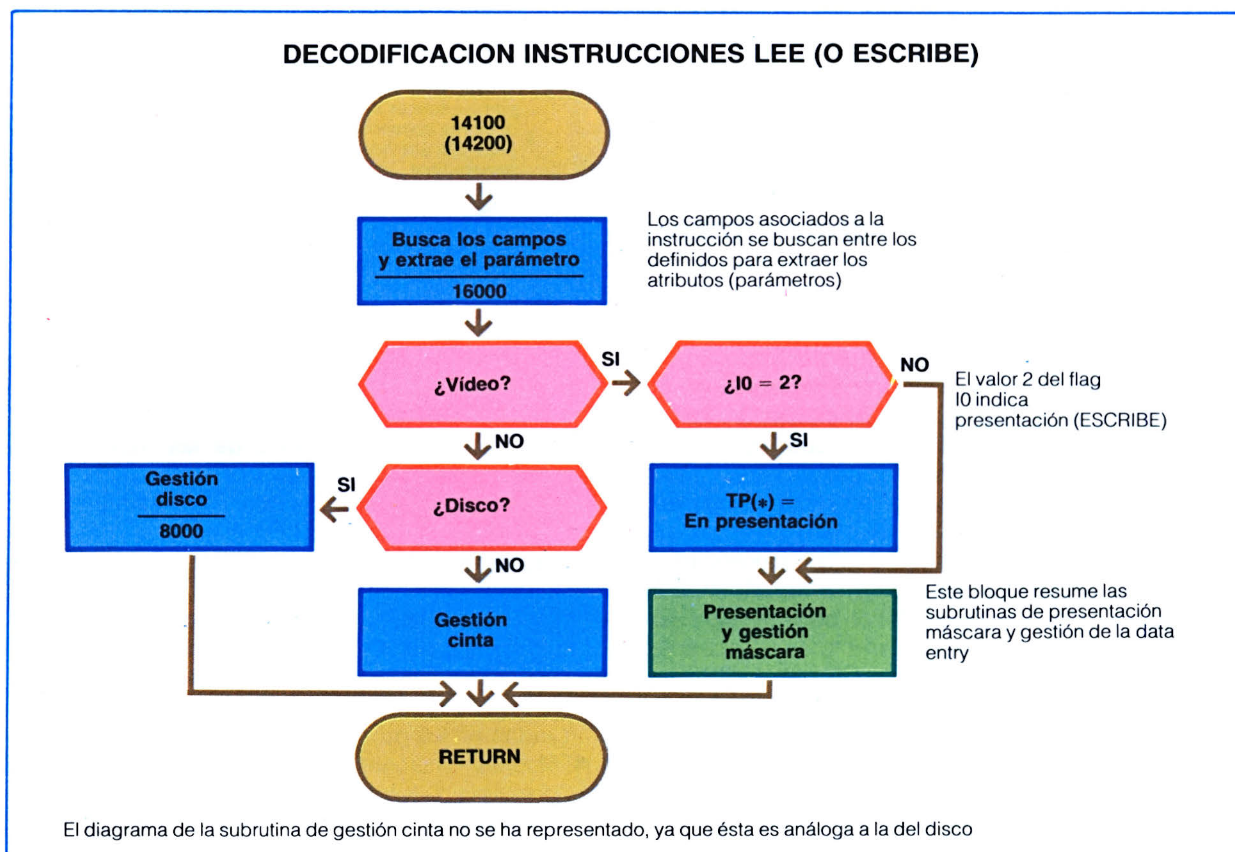
Para incluir la implantación de la impresora basta con un control que determine si el periférico asociado es la impresora, por ejemplo comprobando la sigla «P» (recuérdese que los periféricos, aunque se escriben con su nombre entero, son identificados sólo por la primera letra: V = vídeo, D = disco, C = cinta). Al satisfacerse la condición, hay que llamar una subrutina que envíe a impresión, por ejemplo con una serie de instrucciones LPRINT para la versión MSX, el buffer B$(*), cuyo nombre está especificado en las instrucciones.

Gestión disco. En la página siguiente se ha representado el diagrama de esta subrutina. Los parámetros a implantar antes de la llamada son:

I0 = 1 o 2, según si se desea leer o escribir en el disco.

RE = Número del registro a leer, por lo que sólo se utiliza con I0 = 1

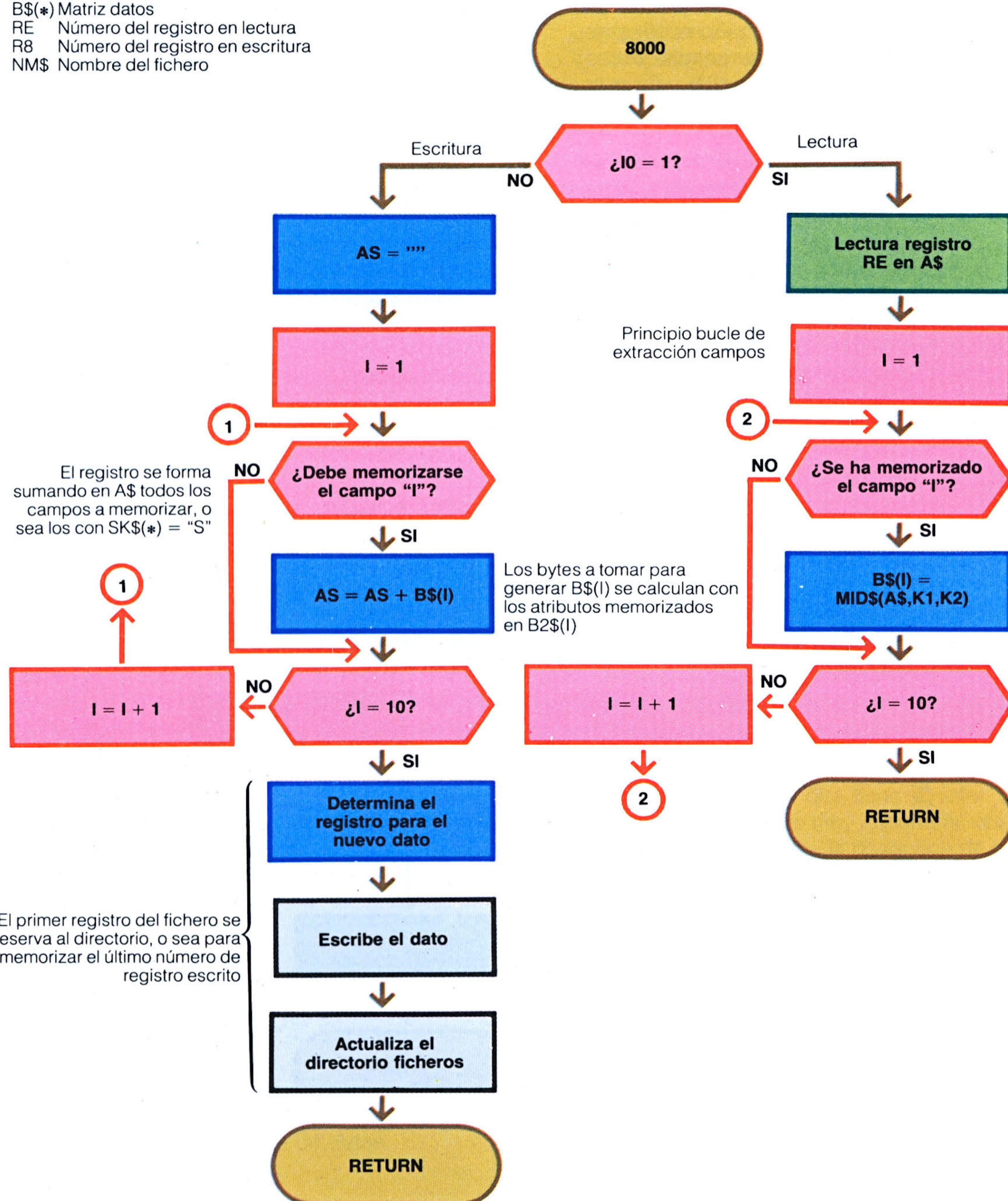
R8 = Número del registro en el que escribir. En esta versión, R8 no debe implantarse en la llamada, porque lo pide el sistema a través del contenido del directorio (registro 1 del fichero; en introducción, el número del re-



El diagrama de la subrutina de gestión cinta no se ha representado, ya que ésta es análoga a la del disco

GESTION DISCO

En la llamada hay que implantar
I0 = 1 si lectura, 2 si escritura
B$(*) Matriz datos
RE Número del registro en lectura
R8 Número del registro en escritura
NM$ Nombre del fichero



La subrutina de gestión cinta es análoga, con las diferencias:
– El fichero siempre se carga interiormente en la memoria
– No hay el directorio

gistro en que escribir se obtiene sumando 1 al valor memorizado en el directorio). Sin embargo, puede considerarse como un parámetro de entrada a fines de futuras implantaciones

NM$ = Nombre del fichero. En esta versión, el nombre del fichero es único y definido *a priori* (definiciones, fase 2), pero puede preverse una implantación que permita el uso de más ficheros simultáneamente. En este caso, los distintos nombres deberán memorizarse en una variable de cadena dimensionada de la que, cada vez, se tomará el nombre el que se desea utilizar, transfiriéndolo seguidamente a NM$.

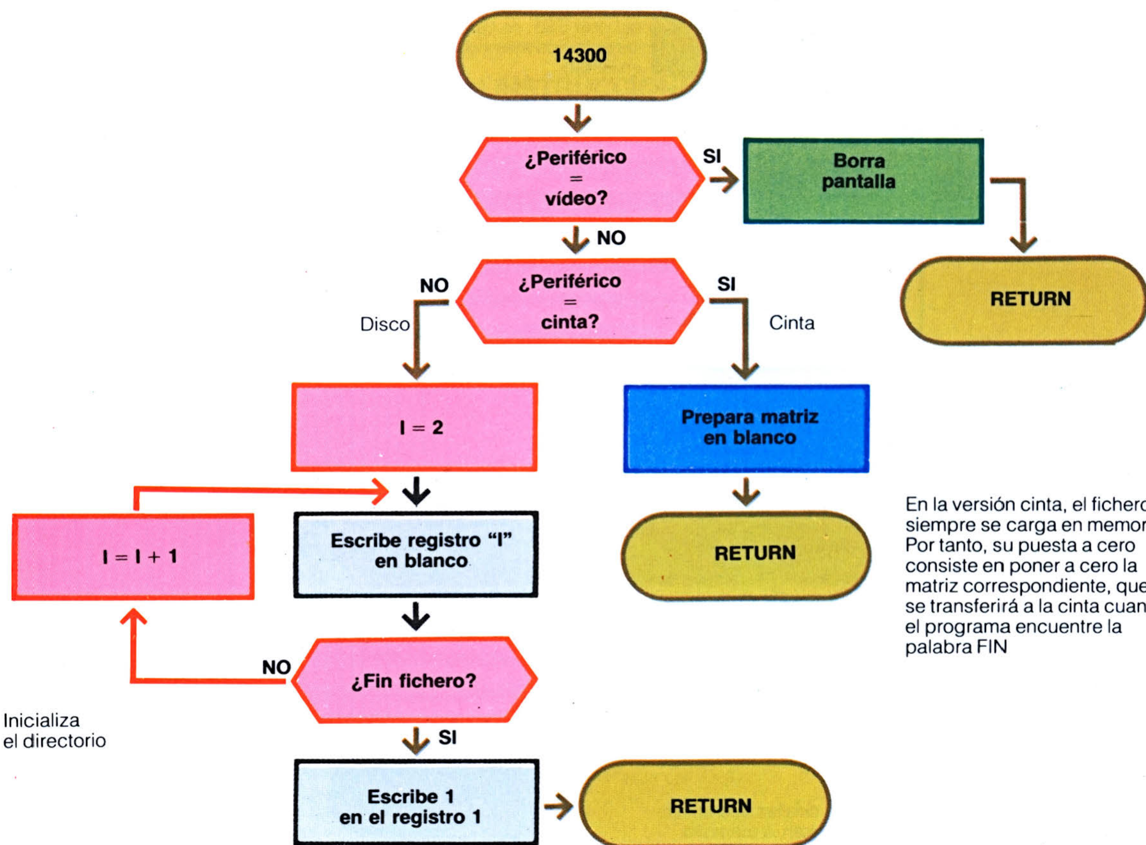
La subrutina, aunque tiene una estructura básica análoga a las utilizadas en los otros programas, presenta algunas diferencias debidas a la parametrización. Esto sucede sobre todo durante la formación de un registro, obtenida con un bucle que suma los contenidos de B$(*); por tanto, hay que tener en cuenta dos casos posibles:

1 / Un campo presente en B$(*) no debe transferirse al fichero

2 / Un campo previsto en el registro, en fase de definición, no está en el buffer B$(*).

El primer caso se supera incluyendo un control en el bucle. Si el campo no es para sumar la instrucción que sigue (suma de los campos A$) se salta. La indicación de si sumar o no el campo deriva de las elecciones realizadas en la fase de definiciones (segunda parte). El otro caso puede introducir errores. Utilizando el programa en generación, cada uno de los campos previstos se inicializa en la longitud justa con espacios en blanco, por lo que basta llamar cada campo en una operación de I/0 para tener la certidumbre de la formación correcta del registro. Sin embargo, debe prestarse atención a eventuales programas sucesivos en los cuales no se utilizan todos los campos que, sin embargo, deben estar presentes.

Instrucción BORRA. Esta instrucción, cuyo diagrama puede verse abajo, permite borrar el contenido del vídeo o el de un fichero.



En el primer caso sólo es una «limpieza de la pantalla», porque el buffer (B$(*)) no se altera. Por tanto, se reduce únicamente a la instrucción de borrado de la pantalla cuyos códigos, en las distintas versiones, deben consultarse en los listados.
En el segundo caso, direccionado a la unidad de disco o cinta, activa en cambio un verdadero borrado de los registros: la lógica seguida es escribir, en todo el fichero, espacios en blanco. Como alternativa se podría activar la remoción total del fichero (con instrucciones Basic del tipo KILL...), solución que, sin embargo, podría ser causa de inconvenientes, como por ejemplo una excesiva fragmentación de los espacios disponibles en el diskette.
La solución elegida presupone que el usuario no desea renovar el fichero, sino sólo borrar su contenido conservando el espacio ocupado. Esta solución también se ha elegido considerando el hecho de que un eventual borrado del fichero puede obtenerse muy fácilmente con los comandos Basic normales o de sistema operativo y, por tanto, que no es necesario preverla en las instrucciones de alto nivel.
Volviendo al diagrama de flujo de la página anterior, las funciones a realizar no necesitan otros comentarios, a no ser la puntualización de que antes de los bucles de escritura de los registros en blanco hay que preparar una cadena que contiene los espacios, cuya longitud debe ser igual a la del registro, o sea a la suma de las longitudes de cada campo que lo constituye. El bucle de preparación de esta cadena (no indicado en el diagrama) debe incluir un control sobre la presencia o no de los distintos campos del registro, o sea un IF... que compruebe la existencia del carácter «S» asociado al campo (ver definiciones, parte 2).

Instrucción BUSCA. La instrucción BUSCA identifica, entre los registros del fichero, el que tiene un determinado valor en el campo especificado. Para analizar el diagrama, en primer lugar deben puntualizarse las modalidades operativas previstas. La instrucción puede utilizarse de tres maneras:

BUSCA nombre de campo
BUSCA nombre de campo <operador> valor
BUSCA nombre de campo <operador> nombre de campo 2

Los operadores previstos son: >, <, =, > =, < =, < >. Por ejemplo, considérese la estructura de un listín telefónico, con los campos de datos:

Apellido
Nombre
Teléfono
Ciudad

La primera forma, por ejemplo con el campo Apellido, se convierte en:

BUSCA APELLIDO

El programa, bajo esta instrucción, busca en el fichero datos el primer registro que en los bytes correspondientes al campo Apellido contiene el mismo valor (alfanumérico) que el contenido en el campo. Por ejemplo, si en la fase de definición se ha asignado a este campo la longitud de 20 bytes, por ser el primer registro, empieza por el byte 1 hasta el byte 20. El programa lee los diferentes registros, extrae la parte interesante (del byte 1 al 20) y la compara con el contenido del campo. Naturalmente, este campo debe tener un valor, el cual debe haberse leído. Generalmente, esta forma se emplea en la búsqueda de un dato leído en pantalla, por lo que la secuencia normal de las instrucciones es:

LEE VIDEO APELLIDO
BUSCA NOMBRE

En cambio, la segunda forma se utiliza para buscar en el fichero el primer registro que en el campo especificado tiene un valor definido con una constante.
Supongamos por ejemplo que el símbolo «*» en el campo Ciudad tenga un significado particular (registro borrado, en suspenso o cualquier otra característica); para obtener la extracción del registro que tiene este indicador, la instrucción será:

BUSCA CIUDAD = «*»

El valor, constante, de referencia puede ser también numérico, como por ejemplo la instrucción:

BUSCA TELEFONO = 12345

Selecciona el registro que tiene el valor 12345 en el campo TELEFONO. Es muy importante observar la diferencia que existe entre la búsqueda de caracteres o la de valores numéricos. El número de teléfono, al no tener que utilizarse para cálculos, también podría definirse de tipo alfanumérico, y en este caso sería necesario respetar sus bordes en cuanto a eventuales espacios en blanco (antes y después del número) que se toman en consideración, y viceversa, definiendo el campo como numérico no tienen ningún peso.
La última forma permite utilizar, como elemento de comparación, el contenido de otro campo, perteneciente o no a la estructura de los datos definida en el fichero. Por ejemplo, la instrucción:

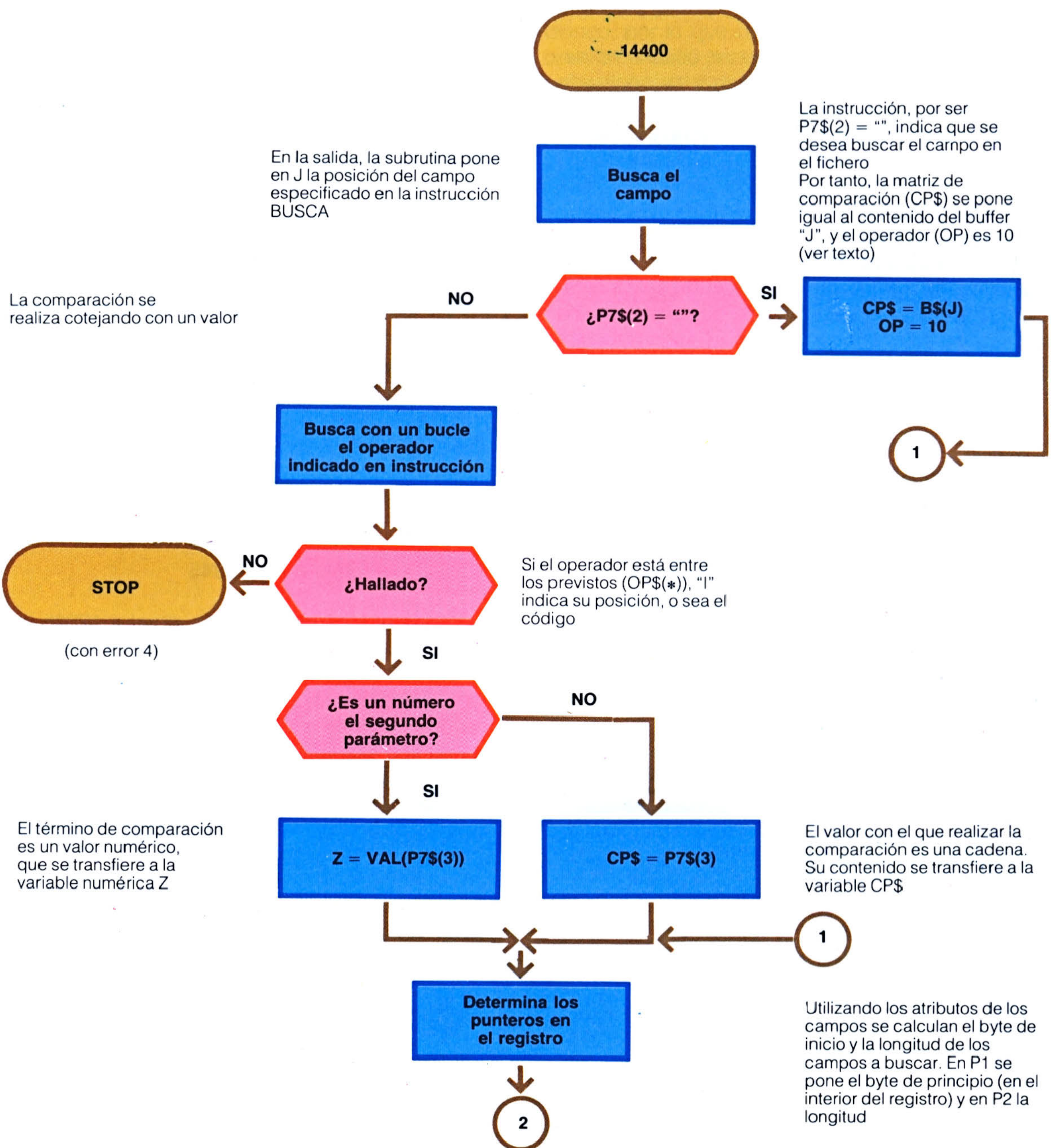
BUSCA APELLIDO = CIUDAD

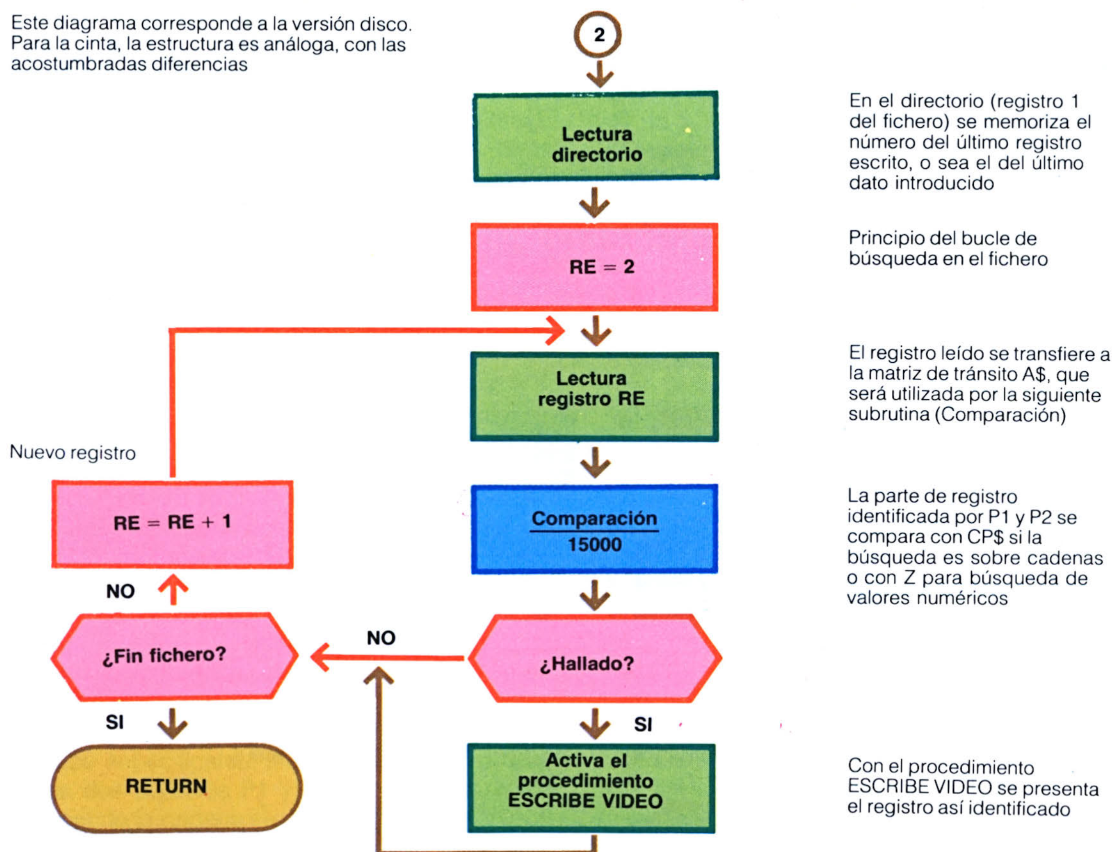
extrae el registro en el que el contenido del campo APELLIDO es igual al de CIUDAD. Como ya se ha indicado, en esta forma también puede utilizarse un campo que no pertenezca al fichero, aunque definido (por tanto, introducido en la fase 1 de las definiciones

DECODIFICACION INSTRUCCION BUSCA (PRIMERA PARTE)

En la llamada debe suministrarse:

OP$(*): En las últimas 6 posiciones contiene los operadores previstos (>, <, > =, =, < =, < >)
P7$(*): Contiene los parámetros
SK$(*): Contiene la letra "S" en correspondencia con los campos que constituyen el fichero





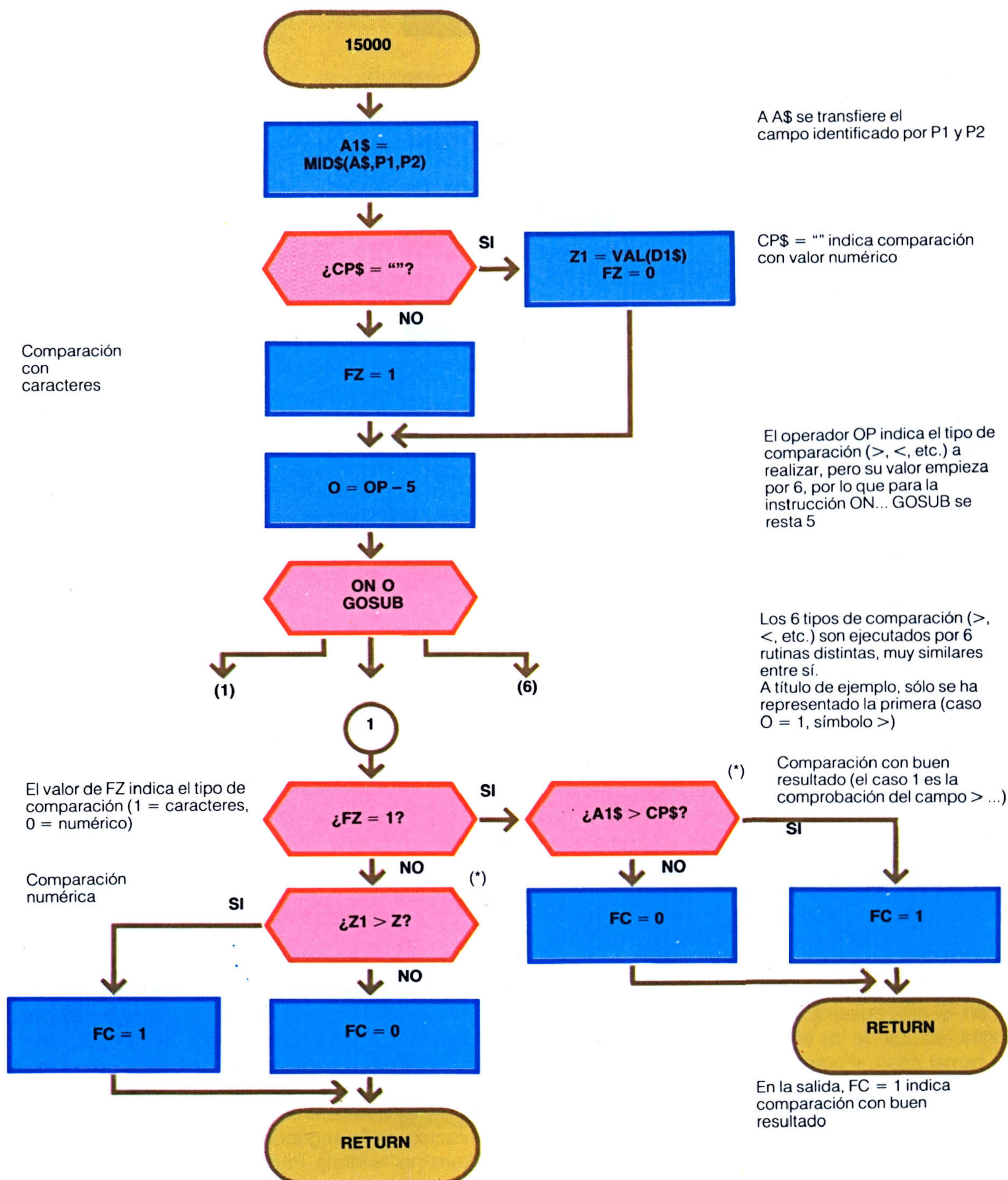
y excluido de la 2, la cual corresponde al formato de registro).
El diagrama de la subrutina puede verse en la página anterior y en la misma figura se han representado los principales parámetros utilizados que deben implantarse antes de la llamada. En la entrada de la subrutina debe determinarse la posición del campo, por lo que se llama una segunda rutina (Busca un campo, ver pág. 2364, que en la salida pone en la variable J la posición del campo seleccionado (por ejemplo, en la estructura de datos anterior, buscando el APELLIDO se tendría J = 1). Este valor se utiliza como puntero para la extracción de los atributos (longitud, tipo, etc.).
Después se comprueba el segundo término de la instrucción. Si está ausente, se trata de la primera forma, y si está, es una de las otras formas.
En el primer caso, el contenido del campo seleccionado, o sea B$(J), se transfiere a la variable CP$ (utilizada para las comparaciones) y el operador que expresa el tipo de comparación a realizar se hace igual a 10 (o sea condición de igualdad). Estas transferencias de contenido entre variables necesitan una explicación.
La subrutina que comprueba la condición implantada en la instrucción es única y, por tanto, debe utilizar dos variables predeterminadas (ver la figura de arriba). En particular controla si entre las variables CP$ y A1$ existe la condición impuesta (>, <, =, etc.).
Esta metodología implica la necesidad de transferir cada vez el valor adecuado a las dos variables:

- la primera debe contener el valor del campo elegido, o sea CP$ = B$(J)
- la segunda (A1$) la implanta el sistema extrayendo de los distintos registros los bytes correspondientes al campo a seleccionar y eventualmente convirtiéndolos en un valor numérico.

La segunda modalidad de funcionamiento, correspondiente a las otras dos formas de la instrucción en las que aparece tanto el operador como el término de comparación, es mucho más compleja. Ante todo hay que comprobar que el operador esté entre los previstos, y si el resultado es positivo, se pasa al análisis de lo escrito en la segunda parte de la instrucción para determinar la forma. Para mayor sencillez, en el diagrama se representa la lógica que corresponde sólo al caso de comparación con una constante (numérica o alfanumérica). En el otro caso (comparación con otro

COMPARACION

En la llamada:
A$: Contiene un registro en el que realizar la comparación
CP$: Valor a buscar, en el caso de búsqueda de cadena
Z : Valor a buscar, en el caso de búsqueda numérica
OP : Código del operador (6 para >, 7 <, 8 > =, 9 < =, 10 =, 11 < >)
P1,P2 : Punteros a A$ para la extracción del valor



(*) Para todos los demás casos sólo varían en consecuencia estas dos instrucciones. Por ejemplo, en el caso O = 2 (OP = 7, <), se hacen A1$ < CP$ y Z1 < Z

campo) hay que memorizar el contenido del campo indicado por P7$(3) y, por tanto, determinar la posición correspondiente al campo cuyo nombre está en P7$(3), por ejemplo K, y extraer y transferir B$(K), poniendo CP$ = B$(K).
A esta fase sigue la determinación de la posición del campo a buscar (P1 y P2) y el bucle sobre el fichero con análisis (comparación) de cada registro. Si la comparación (ver gráfico de la pág. anterior) es positiva, el registro se presenta en pantalla.

Comparación. Los parámetros a implantar antes de la llamada a esta subrutina son:

A$	= Cadena que contiene el dato a analizar
CP$	= Valor a buscar en el caso de comparación con caracteres (cadenas)
Z	= Valor a buscar en el caso de comparación entre valores numéricos
OP	= Indice del tipo de comparación a realizar
P1, P2	= Punteros que identifican, en el interior de A$, la zona a extraer para la comparación.

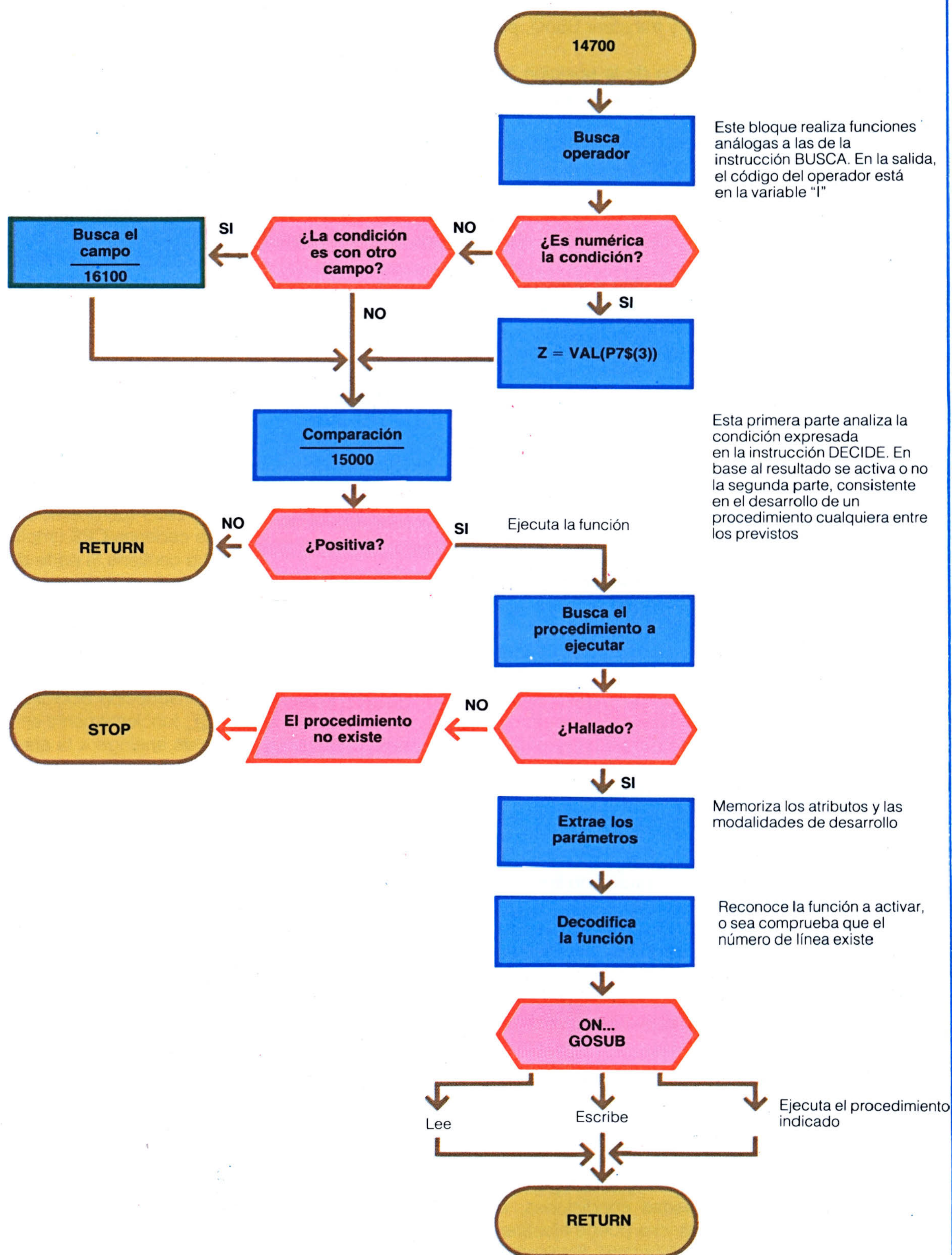
La cadena A$ contiene un registro del fichero que se desea examinar, pero puede utilizarse de cualquier otro modo, por ejemplo transfiriéndole el contenido de una variable (búsqueda en memoria).
Después de esta transferencia (del fichero o de la memoria) debe especificarse, implantando al valor adecuado los dos punteros P1 y P2, en qué elementos (caracteres) de la cadena se desea el control. P1 indica el byte de inicio y P2 el número de bytes a considerar. Por ejemplo, poniendo P1 = 3 y P2 = 5, la comprobación se realiza examinando la parte de A$ comprendida entre el byte 3 y el byte 7 (a partir de 3 y para 5 caracteres, el campo se extrae con la instrucción MID$...). Definidos estos punteros y después de haber cargado A$, hay que especificar el tipo de análisis a realizar. Los operadores previstos son los operadores lógicos normales utilizados en Basic: >, <, > =, < =, =, < >. Para su identificación en el interior de la rutina, el operador OP debe implantarse al valor numérico correspondiente a la posición del operador elegido, teniendo en cuenta que, por motivos estructurales del programa (véanse las subrutinas anteriores), el primer indicador ocupa la posición 6. Por tanto, si se quiere activar la operación «>», hay que implantar OP = 6.
Por último, como la rutina puede comparar tanto cadenas como valores numéricos, en el primer caso hay que transferir la cadena a buscar a CP$, dejando la variable numérica Z al valor 0. En el segundo caso, CP$ debe ser nula, o sea vacía, y Z implantada al valor numérico a buscar.
A los fines de búsqueda únicamente, esta lógica no sería necesaria, porque en ciertas condiciones, la comparación entre valores numéricos puede efectuarse también entre las cadenas equivalentes. Es decir, si la búsqueda debe hacerse en campos con contenido numérico, se podrían comparar los caracteres que contienen. Esta segunda solución, que no utiliza campos numéricos, es mucho más sencilla para la programación, pero también mucho más limitativa.
Por ejemplo, supongamos que en el interior de los registros de un fichero se ha definido un campo numérico de 5 caracteres de longitud y que se desea buscar el registro que contiene el campo numérico 1. Con la comparación por cadenas es necesario generar la variable CP$ de longitud 5 con los primeros 4 bytes en blanco si el valor numérico, como normalmente sucede, está situado a la derecha. En cambio, en el caso de búsqueda en campos númericos, basta con implantar Z = 1, sin tener en cuenta la longitud del campo en el fichero, por lo que es una solución mucho más generalizada y fácil de utilizar.
Sin embargo, este método comporta algunas complicaciones en la escritura de las subrutinas, porque para cada tipo de comparación a realizar hay que prever los dos casos. El problema se ha resuelto implantando un flag que indica, con su valor, el tipo de análisis. Si la comparación es numérica, se comparan las dos variables Z (implantadas en la llamada) y Z1 obtenida convirtiendo en numérica la parte de A$ definida por los punteros P1 y P2; en el otro caso (comparación por caracteres) se comparan las cadenas CP$ (implantadas en la llamada) y A1$, que contiene la parte de A$ sometida a control. Por brevedad, en el diagrama sólo se ha representado la lógica correspondiente al operador «>». Para los demás sirve la misma, excepto la instrucción de control, que varía en función del operador elegido. Finalmente, obsérvese que esta rutina, además de utilizarse para la instrucción BUSCA, también se utiliza para la DECIDE; funcionalmente es diferente, pero estructuralmente es análoga a la anterior.

Instrucción DECIDE. Esta instrucción, cuyo diagrama aparece en la página siguiente, tiene un desarrollo análogo a la BUSCA. Antes de comentar el diagrama hay que llamar la atención sobre su sintaxis y finalidad. La instrucción se introduce en la forma (a menudo la forma no corresponde al formato):

DECIDE

< valor > < operador > { < valor > | < campo 2 > }
< instrucción >

con el siguiente significado: si la condición expresada por < operador > entre < campo > y < valor > (o < campo > y < campo 2 >) es verdadera, entonces ejecuta la instrucción cuyo número está en < instrucción >. Por tanto, el procedimiento consiste en efectuar una comparación (entre un campo y un valor, o entre dos campos) y, si la comparación es positiva, de activar el procedimiento definido en el campo < instrucción >.
Por tanto, la primera función a ejecutar es la decodifi-



DECODIFICACION INSTRUCCION DECIDE
14700
Busca operador
Este bloque realiza funciones análogas a las de la instrucción BUSCA. En la salida, el código del operador está en la variable "I"
¿Es numérica la condición?
NO
¿La condición es con otro campo?
SI
Busca el campo
16100
NO
SI
Z = VAL(P7$(3))
Comparación
15000
Esta primera parte analiza la condición expresada en la instrucción DECIDE. En base al resultado se activa o no la segunda parte, consistente en el desarrollo de un procedimiento cualquiera entre los previstos
¿Positiva?
NO
RETURN
SI
Ejecuta la función
Busca el procedimiento a ejecutar
¿Hallado?
NO
El procedimiento no existe
STOP
SI
Extrae los parámetros
Memoriza los atributos y las modalidades de desarrollo
Decodifica la función
Reconoce la función a activar, o sea comprueba que el número de línea existe
ON... GOSUB
Lee
Escribe
Ejecuta el procedimiento indicado
RETURN

cación del operador que expresa la condición a comprobar (>, <, etc.) a la que le sigue la comparación realizada utilizando la subrutina anterior. Naturalmente, si la comparación es con otro campo, primero hay que buscarlo, o sea determinar su posición, para poder extraer de él el contenido (a utilizar en la comparación). Esta comparación la realiza la subrutina de la página 2364 (búsqueda de un campo). Si la comparación da resultado negativo, la subrutina termina; en caso de que no sea así, se activa la segunda parte, relativa al desarrollo del procedimiento identificado en el campo < instrucción >.
También en este caso, la primera función a ejecutar es la búsqueda del procedimiento, que por un error de introducción podría no existir (recuérdese que, en esta versión, el procedimiento, o sea las instrucciones, se identifica con un número progresivo).
Si la instrucción se encuentra, o mejor dicho, si existe el número de línea especificado, se extraen los parámetros y la instrucción se ejecuta como si se hubiese llamado explícitamente.
Por consiguiente, en esta subrutina hay un bloque de decodificación y uno de selección análogos a los utilizados para enviar a ejecución las distintas instrucciones del programa.

Instrucción REPITE. La instrucción REPITE (ver pág. siguiente) es la equivalente a los bucles en Basic. Su sintaxis es:

REPITE < número de veces >
< procedimiento >
....
< procedimiento >
FIN

El desarrollo consiste en un bucle, repetido el número de veces especificado, que llama en secuencia las instrucciones identificadas por el correspondiente número de línea en los campos < procedimiento >. Los procedimientos a repetir pueden ser un número cualquiera (compatible con la ocupación de memoria); el bucle termina al encontrar la palabra FIN.
Obsérvese que con esta estructura, mantenida voluntariamente sencilla, no puede utilizarse un bucle dentro de otro, o sea la instrucción REPITE no puede referirse a sí misma. Para implantar una función como ésta (análoga de los bucles anidados del Basic), ante todo hay que definir, para cada palabra FIN que se encuentra, cuál es la correspondiente instrucción REPITE (de manera análoga a la función que ejecuta la instrucción NEXT del Basic, que especificando el nombre del índice a incrementar, proporciona el medio para identificar cada bucle).
El diagrama no presenta dificultades particulares, puesto que se trata de una utilización diferente de las lógicas y de las subrutinas ya vistas.
El único punto que es necesario ilustrar es la determinación del número de veces que el bucle debe repetirse. Se han previsto dos casos:

- el número es expresado con un valor, que simplemente debe extraerse (N = VAL...)
- el número lo proporciona el contenido de un campo. En este caso, primero hay que buscar el campo (definido con el nombre en < número de veces >) y de éste extraer el valor.

Después de esta función puede activarse el bucle que consiste en llamar todos los procedimientos especificados hasta la palabra FIN.
Obsérvese que el diagrama sólo es de principio, y muestra la lógica del bucle indicando cómo se realiza un solo procedimiento. En realidad, el listado contiene una parte de más, que es precisamente el desarrollo de todos los procedimientos comprendidos entre la instrucción REPITE y la palabra FIN.

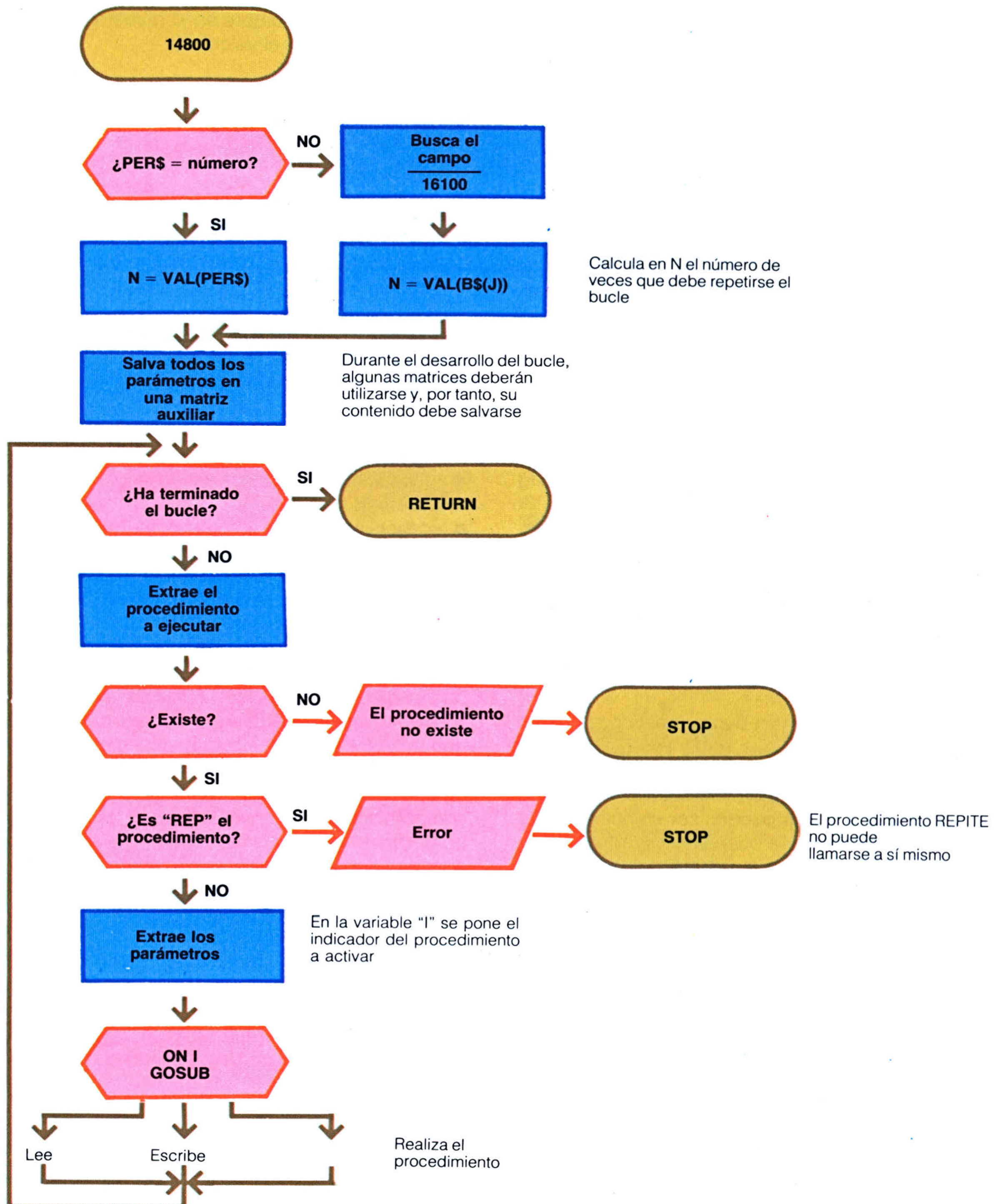
Instrucción CALCULA. El desarrollo de una fórmula puede sintetizarse como sigue (en la hipótesis de una sola operación, y por tanto dos operandos):

- reconocimiento y extracción del primer operando
- extracción del operador
- reconocimiento y extracción del segundo operando
- desarrollo del cálculo
- conversión en cadena del resultado.

La primera función (reconocimiento del operando) necesita un análisis sobre el tipo de dato. Efectivamente, el operando puede ser un valor numérico, en cuyo caso sólo debe transferirse a una variable numérica, o bien el nombre de un campo. En este caso hay que buscarlo entre los definidos y, si está, extraer su valor. Sólo después de este paso se dispone del valor numérico a utilizar para el cálculo (también en este caso a convertir en cadena).
Siguen la búsqueda del operador con un bucle de comparación entre el presente en la operación y los operadores previstos en los DATA, y la extracción del segundo operando, realizada de manera análoga a la primera.
Terminadas las extracciones, se tiene:

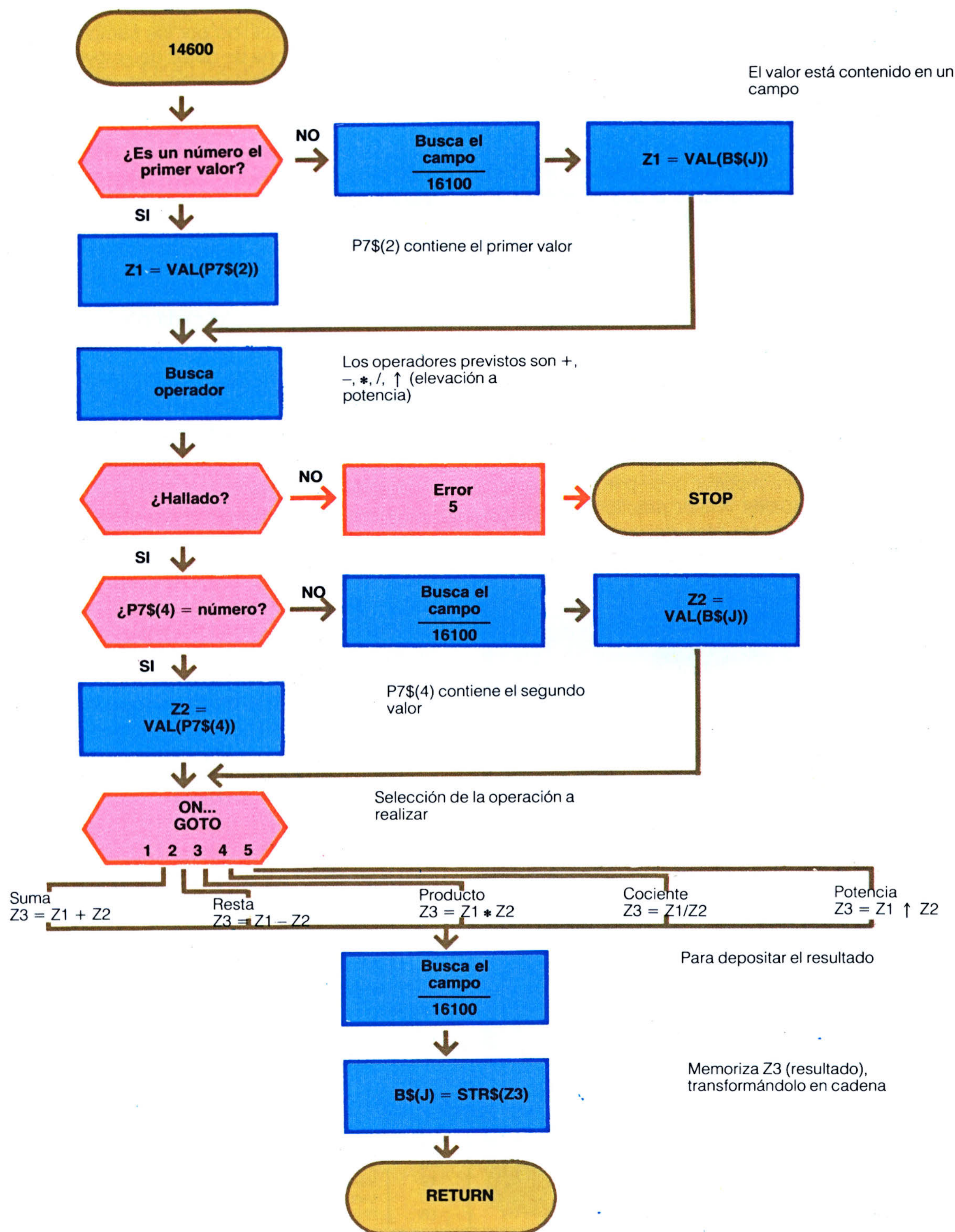
- Z1 = Valor numérico correspondiente al primer operando
- Un flag, fijado en la búsqueda del operador, que indica el cálculo a realizar
- Z2 = Valor numérico del segundo operando.

Sólo queda por activar la línea de programa (o la subrutina) correspondiente a la operación, o sea la línea apuntada por el flag.
Realizada la operación, el resultado que se encuentra en la variable Z3 se transforma en cadena y se transfiere al campo indicado en la operación.


DECODIFICACION DE REPITE
En la llamada, PER$ contiene el número de veces que debe repetirse el bucle, o el nombre del campo que, a su vez, contiene este valor
14800
¿PER$ = número?
NO
Busca el campo
16100
SI
N = VAL(PER$)
N = VAL(B$(J))
Calcula en N el número de veces que debe repetirse el bucle
Salva todos los parámetros en una matriz auxiliar
Durante el desarrollo del bucle, algunas matrices deberán utilizarse y, por tanto, su contenido debe salvarse
¿Ha terminado el bucle?
SI
RETURN
NO
Extrae el procedimiento a ejecutar
¿Existe?
NO
El procedimiento no existe
STOP
SI
¿Es "REP" el procedimiento?
SI
Error
STOP
El procedimiento REPITE no puede llamarse a sí mismo
NO
Extrae los parámetros
En la variable "I" se pone el indicador del procedimiento a activar
ON I GOSUB
Lee
Escribe
Realiza el procedimiento

DECODIFICACION DE CALCULA

El diagrama presentado, por motivos de sencillez, se refiere a la subrutina en la forma más elemental, y sólo puede realizar cálculos de tipo Resultado = Valor 1(+, −, *, /, ↑) valor 2



Ejemplos de aplicación

Para ilustrar la técnica particular que debe seguirse al escribir programas con este lenguaje, a continuación se indican algunos ejemplos de aplicación.
Para todos ellos deberán tenerse siempre presentes las principales reglas:

1 / Antes de utilizar cualquier campo, debe definirse, y dado que esta función precede a la de introducción de las instrucciones, hay que definir *a priori* todos los campos que se quieren utilizar, distinguiendo entre los que deben transferirse al fichero y los destinados únicamente al uso en memoria
2 / La longitud del fichero datos debe definirse *a priori*, por lo que siempre debe tenerse por lo menos una idea del orden de magnitud de las cantidades de datos a memorizar
3 / La fase de las definiciones no admite correcciones, por lo que hay que prestar mucha atención en su preparación, bajo pena de tener que volver a escribir los atributos de todos los campos.

Programa con actualización de datos. El programa que sigue no está identificado con una aplicación concreta, pero sirve para mostrar el empleo de todas las instrucciones principales. A fines prácticos puede aplicarse a la gestión de un plan de cuentas o para el control de compras a plazos si a los campos numéricos se les atribuye el significado de dinero, o bien como gestión de almacén si los campos tienen el significado de cantidad.
Considérese la siguiente máscara:

```
Código        –
Descripción   ––––––––––––––––––––
Importe       –––––––
Saldo         –––––––
Movimiento    –––––––
Resto         –––––––
```

Identifican una estructura de datos dividida en tres tipos:

- El primero, constituido por los campos Código y Descripción, es una parte descriptiva utilizada para identificar un determinado elemento. En la aplicación al plan de cuentas puede ser el código de la cuenta, en las compras el objeto, y en el almacén la voz, o sea el artículo
- Siguen los campos Importe y Saldo (en la aplicación de almacén podrían ser Existencia inicial y Existencia actual)
- El último, constituido por los campos Movimiento y Resto, es la zona de datos de «movimientos», o sea de los valores que deben sumarse o restarse para obtener el valor actual de balance.

Para fijar las ideas, supongamos que con esta aplicación se gestiona un recordatorio de las situaciones de pagos a plazos.
El Código y la Descripción, como se ha indicado, identifican el objeto al que se refiere el registro. El importe es la cifra total (costo), mientras que el Saldo constituye la cifra que aún hay que pagar, y se obtiene restando del importe inicial cada pago realizado. Expresado en fórmula:

Saldo = Importe – Suma movimientos

Como los distintos pagos se realizan en momentos sucesivos, por ejemplo cada mes, el mejor modo de calcular el Saldo consiste en atribuirle un valor inicial igual al importe, restando cada movimiento en el momento de su introducción.
Para obtener esta lógica existen otros dos campos: el Movimiento (que es el importe del pago a restar al Saldo) y el Resto, que se utiliza como variable de tránsito.
Además, ni el Movimiento ni el Resto deben transferirse al fichero, que está estructurado para contener solamente los datos descriptivos (Código y Descripción), el valor inicial (Importe) y el actual después de cada movimiento (Saldo).
Como de costumbre, la primera fase es la definición. En ella deben identificarse todos los campos (incluso si no se transfieren al disco).
Por ejemplo, una estructura podría ser la presentada en la tabla de abajo.
Como ya se ha indicado, sólo deben transferirse al fichero los campos CODIGO, DESCR., IMPORTE y SALDO, por lo que en la fase 2 de las definiciones se introducirá la letra «S» sólo en estos campos y no en los otros (MOVIMIENTO y RESTO), que se dedican a la introducción y al cálculo. Como es habitual, la apli-

Nombre	Tipo	Long.	Dec.	Colum.	Línea	
CODIGO	1	1	–	5	5	Identifica el registro
DESCR.	3	20	–	5	7	Descripción de la voz
IMPORTE	1	7	–	5	9	Valor inicial mantenido como recordatorio
SALDO	1	7	–	10	12	Valor que queda después de cada movimiento
MOVIMIENTO	1	7	–	10	15	Valor del movimiento
RESIDUO	1	7	–	10	18	Utilizado como matriz de cálculo

cación se realiza con dos programas. El primero corresponde a la fase de generación de datos para el archivo, y el segundo se dedica a la generación de movimientos, o sea a las actualizaciones y a los cálculos.

Creación y archivo. En la fase de creación, tras las definiciones anteriores, las funciones a realizar son:

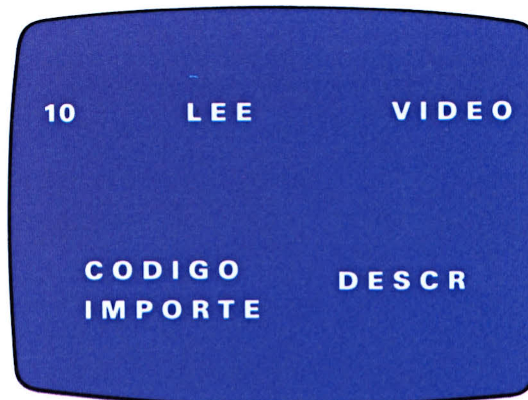
1 / Introducción de CODIGO, DESCR. e IMPORTE
2 / Cálculo del valor inicial del SALDO (inicialmente es igual al IMPORTE, porque no se ha efectuado ningún movimiento)
3 / Memorización en disco.

El conjunto se repite el número de veces necesario, o sea el número de plazos que se desea realizar (como en este ejemplo al campo CODIGO se le ha reservado un solo byte, el número máximo es 9).
El programa correspondiente a las funciones expuestas se ha representado en esta página y en las siguientes.
Las instrucciones son todas sencillas y comprensibles, excepto la 20, que necesita algunos comentarios.
Ante todo hay que recordar la sintaxis de ilustración CALCULA, que no puede prever el signo "=" y, por tanto, se sobreentiende que entre el primer campo, en el que se depositará el resultado, y los siguientes, que expresan el cálculo a realizar, es como si existiese el símbolo "="; por tanto, la instrucción equivale a:

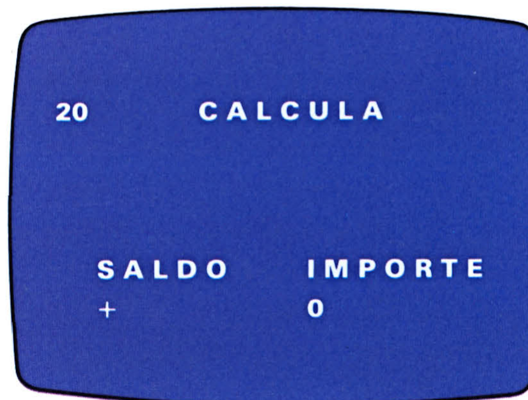
SALDO = IMPORTE + 0

El objeto de esta instrucción es el de transferir al campo SALDO el valor contenido en el campo IMPORTE, o sea: SALDO = IMPORTE.
Esta versión del programa no prevé instrucciones de este tipo, pero necesita dos operandos y un operador, por lo que para ejecutar la función deseada (transferencia de contenido) es necesaria una operación aritmética que deje sin variación el valor, por ejemplo la suma con 0 o equivalente, como por ejemplo el producto o la división por 1.
Por último, obsérvese que el bucle tiene un índice hasta el valor 9, lo que implica la necesidad de introducir 9 registros (el bucle no puede ser interrumpido), pero podría parametrizarse fácilmente, por ejemplo introduciendo antes de la 10 una instrucción de lectura del número de veces de repetición del bucle. Naturalmente, este campo deberá definirse de la misma manera que los otros. La correspondiente instrucción de lectura debe realizarse fuera del bucle y antes de que empiece, por lo que deberá efectuarse en la línea 5.

Para cada dato se leen en pantalla los campos de descripción (CODIGO, DESCR) y el importe.



El SALDO inicial debe ser igual al importe. Se calcula poniendo:
SALDO = IMPORTE + 0
El cálculo + 0 es necesario para respetar la sintaxis de la instrucción.



Todos los campos se han representado, incluyendo el SALDO calculado.

```
30    ESCRIBE    VIDEO

CODIGO           DESCR
IMPORTE          SALDO
```

Los mismos se transfieren al disco.

```
40    ESCRIBE    DISCO

CODIGO           DESCR
IMPORTE          SALDO
```

Con esta instrucción se activa el bucle de repetición para 9 veces. Así se introducen 9 grupos de datos (registros) diferentes.
El bucle se ejecuta sobre las instrucciones 10, 20, 30 y 40. La palabra FIN indica el final del bucle y no el del programa.

```
60        REPITE        9

10                     20
30                     40
FIN
```

Para dar a la palabra FIN el significado de final del programa, hay que escribirla en el campo reservado a la instrucción, o sea en la misma línea del número de línea (que debe ser 99).

```
99        FIN
```

```
10 REM ******************************
11 REM *                            *
12 REM * ORDENADOR PARA EL ORDENADOR *
13 REM *           C - 64           *
14 REM *                            *
20 REM ******************************
21 DIM TF(15),D$(25),VO$(20),FR$(8)
22 DIM B2$(60),IST$(15),S0(15),NT(15),PER$(15),PAR$(15,10),OP$(11)
24 BL$=" ":PD$=".":AS$="-":NS$="#":SU$="▌":GIU$="▟":HO$="▜":SL$="/"
25 US$=CHR$(133):NV=5:XC=12:YC=6:SZ=2:G9=0:POKE 650,128
30 GOSUB20000
35 FOR I=1 TO 15:READ TF(I):NEXTI
40 FOR I=1 TO 12:READ H$(I):NEXTI
50 FOR I=1 TO 11:READ OP$(I):NEXTI
55 FOR I=1 TO 8:READ FR$(I),NP(I):NEXTI
60 M3=10
65 A1=32:A2=95
80 PRINTHO$:GOSUB1300:PRINTHO$:PRINTCHR$(142):REM ** MENU **
90 IF OPZ=5 THEN PRINTHO$:POKE646,6:END
110 IF G5>1 AND OPZ=2 THEN GOSUB 710:GOTO150
120 ON OPZ GOSUB 12000,700,990,14000
150 GOTO 80
700 REM *** FASE 2,INSTRUCCIONES ***
705 G5=1
710 FOR NI=G5 TO M3
712 OZ=3:GOSUB7000
714 IN=CY(4)
716 FOR I=5 TO 8:TP(I)=TP(4):LC(I)=LC(4):ND(I)=ND(4):CX(I)=CX(4)
718 IN=IN+1:CY(I)=IN:NEXT I
720 IN=CY(4)-1
722 FOR I=9 TO 13:TP(I)=TP(4):LC(I)=LC(4):ND(I)=ND(4)
724 CX(I)=CX(4)+18:IN=IN+1:CY(I)=IN:NEXT I
728 NUC=13
730 GOSUB 2000
732 J=1
735 GOSUB 3040
740 IF FS=6 THEN 730
742 IF FS=7 THEN 750
744 IF FS=9 THEN 735
746 GOTO 760
750 IF NI>1 THEN 950
752 RETURN
760 IF LEFT$(B$(2),3)="   "ORLEFT$(B$(2),3)="###"THENSI=9:GOTO780
765 FOR K=1 TO 8:IF LEFT$(B$(2),3)=FR$(K) THEN S0(NI)=K:SI=K:K=8:GOTO780
770 NEXT K
775 PRINTHO$:X=17:Y=15:GOSUB11000:PRINT"‾—┌_"
776 FORT=1TO1000:NEXTT:PRINTHO$:OZ=5:GOSUB7000:GOTO730
778 ERR=0
780 ON SI GOSUB 800,800,830,880,840,880,852,870,860
785 IF ERR<>0 THEN PRINTHO$:OZ=5:GOSUB7000:GOTO730
790 PRINTHO$:GOTO 900
799 REM *********** LEE,ESCRIBE ***********
800 J$=LEFT$(B$(3),1):IFJ$<>"V"ANDJ$<>"D"THEN813
801 IF LEFT$(B$(4),3)="   "ORLEFT$(B$(4),3)="###" THEN 813
802 FOR J=4 TO 3+NP(SI)
803 IF LEFT$(B$(J),3)="   "ORLEFT$(B$(J),3)="###" THEN 815
804 FOR I=1 TO 60 STEP 6
805 IF B2$(I)=B$(J) THEN 810
807 NEXTI:GOTO813
810 NEXTJ:GOTO815
813 PRINTHO$:X=5:Y=10:GOSUB11000:PRINT"‾—┌/┬,REPITA LA INSTRUCCION":FORT=1TO15
00:NEXTT
814 ERR=1:RETURN
815 PRINTHO$:X=15:Y=10:GOSUB11000:PRINT"┤┌—‾┤┌"
819 FOR T=1 TO 1000:NEXTT:ERR=0:RETURN
830 REM ************* BORRA ****************
832 B$(3)=LEFT$(B$(3),1)
833 IF B$(3)="V"ORB$(3)="D"THENGOSUB815:RETURN
835 GOTO 813
840 J=4:REM **************** CALCULA **************
841 FORI=1TO60STEP6:IFB2$(I)=B$(J)THEN843
842 NEXTI:GOTO813
843 IF B2$(I+1)="3" THEN 813
844 IF J=7 THEN 849
845 IFJ=5THENJ=7:GOTO 847
846 J=5
847 IF ASC(B$(J))<48ORASC(B$(J))>57THEN 841
848 GOTO844
849 FORI=1TO5:IFLEFT$(B$(6),1)=OP$(I)THEN851
850 NEXTI:GOTO813
851 GOSUB 815:RETURN
852 REM ******************** REPITE ****************
853 FORI=4TO13
854 IF LEFT$(B$(I),3)="FIN" THEN 859
855 IF VAL(B$(I))=VAL(B$(1))THEN813
856 REM
858 NEXTI:GOTO813
859 GOSUB 815:RETURN
860 REM ********** BORRA INSTRUCCION **************
861 FORI=1TONI:IFNT(I)=VAL(B$(1))THENNT(I)=0:IST$(I)="":B$(1)="0"
862 NEXTI
864 RETURN
870 RETURN
880 REM ********** BUSCA,DECIDE ******************
882 FORI=1TO60STEP6:IFB2$(I)=B$(4)THEN884
883 NEXTI:GOTO813
884 IF (LEFT$(B$(5),3)="   "ORLEFT$(B$(5),3)="###")ANDSI=4THEN898
889 FORI=6TO11:IFLEFT$(B$(5),2)=OP$(I)THEN894
890 NEXTI:GOTO813
894 IF SI=4 THEN 898
895 FORI=1TO99:IFVAL(B$(7))=ITHEN 898
896 NEXTI:GOTO813
898 GOSUB 815:RETURN
900 REM ** FASE DE ASUNCION **
902 IST$(NI)=B$(2):NT(NI)=VAL(B$(1))
903 L$=LEFT$(IST$(NI),3)
905 IF L$=FR$(8) THEN NT(NI)=99:GOTO950
910 PER$(NI)=LEFT$(B$(3),1)
915 IF L$="REP" THEN PER$(NI)=B$(3)+"    "
920 FORI=1TONP(SI):PAR$(NI,I)=B$(I+3)
922 IF LEFT$(PAR$(NI,I),3)="   "ORLEFT$(PAR$(NI,I),3)="###" THEN PAR$(NI,I)=""
925 NEXTI
930 NEXTNI
935 NI=NI-1
950 FOR I=0 TO NI-1
952 IF NT(I)<=NT(I+1) THEN 956
953 N1=NT(I):NT(I)=NT(I+1):NT(I+1)=N1:I1$=IST$(I):IST$(I)=IST$(I+1)
954 IST$(I+1)=I1$:N1=S0(I):S0(I)=S0(I+1):S0(I+1)=N1:GOTO980
956 IFNT(I)<>NT(I+1)THEN960
957 FORJ=ITONI-1:NT(J)=NT(J+1):IST$(J)=IST$(J+1):PER$(J)=PER$(J+1):S0(J)=S0(J+1)
958 FORII=1TO10:PAR$(J,II)=PAR$(J+1,II):NEXTII:NEXTJ:IST$(NI)="     ":NI=NI-1
959 GOTO950
960 NEXTI:G5=NI+1:RETURN
980 I1$=PER$(I):PER$(I)=PER$(I+1):PER$(I+1)=I1$
981 FORII=1TO10:I1$(II)=PAR$(I,II):PAR$(I,II)=PAR$(I+1,II)
982 PAR$(I+1,II)=I1$(II):NEXT II
984 GOTO 950
985 REM ---------------------------------
990 REM ******* LISTADO ********
991 GOSUB 9850:PRINTHO$:I=1:Y=1:IF R=2 THEN OPEN4,4:CMD4
992 PRINT"◣"NT(I);TAB(4)"▝"IST$(I);"  ▝▝";PER$(I):II=1
993 IFPAR$(I,II)=""THEN995
994 PRINTTAB(27)PAR$(I,II):Y=Y+1:II=II+1:IF II<=NP(S0(I)) THEN 993
995 PRINT:PRINT:Y=Y+3
996 IF R=1 AND Y=>20 THEN GOSUB 9800:PRINTHO$:Y=1
997 I=I+1:IF I<=G5-1 THEN 992
998 IF R=2 THEN PRINT#4:CLOSE4:GOTO1000
1000 X=2:Y=24:GOSUB11000:PRINT"▌PULSAR UNA TECLA PARA VOLVER AL MENU";
1005 POKE 198,0:WAIT198,1:POKE198,0
1010 RETURN
1300 REM ****************************
1301 REM *    MENU  PRINCIPAL     *
```

```
1302 REM *****************************
1303 LV=-1:POKE53281,6:POKE53280,14
1305 U$=" \ENU  ":OPZ=0:VO$(OPZ)="  Γ-‾/♠T- PARA EL Γ-‾/♠T-":GOSUB 7000
1309 OPZ=1
1315 SS=53248:POKESS+21,1:POKE2040,13
1316 RESTORE
1318 FOR I=0 TO 62:READW:POKE832+I,W:NEXTI
1321 POKESS+39,1:C1=95:C2=111
1323 GOSUB 1363
1324 GOSUB 1396
1327 GET OPZ$:IF OPZ$="" THEN 1327
1333 IF OPZ$=GIU$ THEN C2=C2+16:OPZ=OPZ+1:GOSUB1363
1336 IF OPZ$=SU$ THEN C2=C2-16:OPZ=OPZ-1:GOSUB1363
1339 IF OPZ$=US$ THEN 1381
1342 IF ASC(OPZ$)=25 THEN GOSUB 1456:GOTO1305
1345 GOTO 1327
1363 REM **
1366 IF C2<111 THEN C2=C2+16:OPZ=OPZ+1:RETURN
1369 IF C2>175 THEN C2=C2-16:OPZ=OPZ-1:RETURN
1372 POKESS,C1:POKESS+1,C2
1375 RETURN
1378 REM **
1381 X=XC:Y=OPZ*SZ+YC
1384 GOSUB 11000
1387 PRINT"▲";VO$(OPZ)
1390 FOR T=1 TO 1000:NEXTT
1392 POKESS+21,PEEK(SS+21)AND254
1393 RETURN
1396 REM
1399 FOR J=1 TO NV:READ VO$(J):NEXTJ
1405 FOR II=1 TO NV
1408 X=XC:Y=II*SZ+YC
1411 GOSUB11000
1414 PRINT"▲";VO$(II)
1417 NEXT II
1420 RETURN
1425 DATA0,0,0,96,0,0,144,0,3,32,0
1427 DATA4,64,0,8,143,252,17,112,2,224,129,252
1429 DATA0,63,0,0,65,0,0,63,0,0,31,0
1431 DATA0,33,0,0,31,0,224,63,0,16,81,0
1433 DATA15,30,0,0,0,0,0,0,0,0,0,0,0,0
1444 DATA"1- ‾EFINICIONES"
1445 DATA"2- ⌐NSTRUCCIONES "
1448 DATA"3- LISTADO "
1452 DATA"4- ‾JECUCION"
1454 DATA"0- _IN DE TAREA"
1456 REM ******** AYUDA MENU **********
1459 PRINTCHR$(147):PRINTCHR$(14)
1462 POKESS,255:POKESS+1,115:POKESS+39,6
1465 POKESS+40,6:POKESS+41,6
1468 POKESS+21,7:POKE2040,13:POKE2041,14:POKE2042,15:POKESS+29,4
1471 FORN=0TO62:READQ:POKE832+N,Q:NEXTN
1474 FORN=0TO62:READQ:POKE896+N,Q:NEXTN
1477 FORN=0TO62:READQ:POKE960+N,Q:NEXTN
1480 POKESS+39,1:POKESS+40,1:POKESS+41,1
1483 POKE 53281,6
1486 PRINT"┌──────────────────┐";
1489 FOR I=1 TO 15:PRINT"│                                        │";:NEXTI
1492 PRINT"└──────────────────┘";
1495 PRINTCHR$(19):PRINTTAB(7)"
1498 POKESS+2,255:POKESS+3,145:POKESS+4,245:POKESS+5,180
1501 PRINT"DESPLAZA ABAJO:"
1504 PRINT"DESPLAZA ARRIBA:"
1507 PRINT"INTRODUCE OPCION ELEGIDA:"
1510 PRINTTAB(3)"
1513 GET R$:IFR$=""THEN1513
1550 POKESS+39,6:PRINT"":RETURN
1552 DATA3,255,192,12,0,32,16,16,16,32,56,8,32,108,8
1554 DATA64,40,4,64,40,4,64,0,4,93,206,228,81,72,164
1556 DATA81,206,228,81,66,164,93,46,148,64,0,4
1558 DATA64,40,4,64,40,4,32,108,8,32,56,8,16,16,16
1560 DATA12,0,32,3,255,192
1562 DATA3,255,192,12,0,32,16,32,16,32,120,8,32,192,8
1564 DATA64,120,4,64,32,4,64,0,4,93,206,228,81,72,164
1566 DATA81,206,228,81,66,164,93,46,148,64,0,4,64,16,4
1568 DATA64,120,4,32,12,8,32,120,8,16,16,16,12,0,32,3,255,192
1570 DATA0,0,0,7,255,224,24,0,24,96,0,6,128,0,1,129,1,129
1572 DATA134,131,129,132,2,129,132,0,129,142,0,129
1574 DATA132,0,129,132,0,129,132,0,129,128,0,1,96,0,6
```

```
1580 DATA24,0,24,7,255,224,0,0,0,0,0,0,0,0,0,0,0
1600 REM FIN DEL MENU PRINCIPAL
2000 REM
2005 REM *** PRESENTA MASCARA ***
2015 FORI=1TODS:X=DX(I):Y=DY(I):IF TP(I)<1ORTP(I)>5THEN2018
2016 GOSUB11000:PRINT"▲"D$(I)
2018 NEXT I
2030 FOR J=1 TO NUC
2031 IF TP(J)=0 THEN 2130
2032 A$=""
2050 IF TP(J)<>5 AND TP(J)<>4 THEN 2060
2054 X=CX(J):Y=CY(J):GOSUB11000:PRINT"▲";B$(J)
2056 GOTO 2130
2060 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J)=2 THEN C$=NS$:GOTO2065
2064 C$=AS$
2065 IF N1=1 THEN A$=C$:GOTO 2095
2070 FOR I=1 TO N1-N2-1:A$=A$+C$:NEXTI
2075 IF N2=0 THEN A$=A$+C$:GOTO 2095
2080 A$=A$+PD$
2090 FOR I=1 TO N2:A$=A$+"0":NEXTI
2095 X=CX(J):Y=CY(J):GOSUB 11000
2096 PRINT "▓";A$;"▲"
2097 B$(J)=""
2098 FOR K=1 TO N1-N2-1
2100 B$(J)=B$(J)+NS$
2105 NEXT K
2106 IF N2=0 THEN B$(J)=B$(J)+NS$:GOTO 2130
2110 B$(J)=B$(J)+PD$
2120 FORK=1TON2:B$(J)=B$(J)+NS$:NEXTK
2130 NEXT J
2150 RETURN
3000 REM
3010 REM ** DATA ENTRY **
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3041 IF TP(NC)=0 THEN 3105
3050 CC=1
3060 GOSUB 3240
3065 IF FS=9 THEN RETURN
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 REM CARACTER
3105 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 FS=0:J=NUC:GOTO 3040
3150 IF J<NUC THEN J=J+1:GOTO 3040
3160 FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB11000:GOTO 3060
3200 FS=0:GOTO 3040
3210 REM
3220 IF FS>8 THEN FS=0:GOTO 3060
3230 IF FS=8 THEN FS=0:GOTO 3100
3235 RETURN
3240 REM
3250 REM  ** LECTURA DE 1 CAMPO **
3280 FS=0
3290 IF TP(NC)=4 OR TP(NC)=0 THEN RETURN
3300 IF TP(NC)=2 THEN GOSUB 9000:RETURN
3310 IF MID$(B$(NC),CC,1)=PD$ THENX=X+S:CC=CC+S
3320 GOSUB 11000
3330 GET A$:IF A$<>"" THEN 3338
3332 GOSUB11000:PRINT"▬";:GOSUB11000:PRINT"▲";MID$(B$(NC),CC,1);
3334 FOR T=1 TO 20:NEXT T
3336 GOTO 3330
3338 IFASC(A$)=19ORASC(A$)=20ORASC(A$)=147ORASC(A$)=148 ORASC(A$)=44 THEN 3330
3340 PRINT"▮";A$;
3350 CA=ASC(A$)
3360 FOR I=1 TO 15
3370 IF TF(I)=CA THEN FS=I:I=15:GOSUB3630:RETURN
3375 IF FS=9 THEN RETURN
3380 NEXTI
3385 IF CA<A1 OR CA>A2 THEN 3330
3390 IF TP(NC)<>1 THEN 3450
3400 IF CA=45 OR CA=46 OR(CA>=48ANDCA<=57)THEN3420
```

```
3410 GOTO 3330
3420 IF CA<>46 THEN3450
3430 GOSUB 4270
3440 IF SWP=1 THEN 3410
3450 S=1:GOSUB 3520
3460 IF CC=LC(NC) THEN RETURN
3470 CC=CC+1:X=X+1:GOTO 3310
3520 REM
3521 REM ** TRASFERENCIA AL BUFFER **
3540 REM
3570 IF CC=1 THEN SN$="":GOTO 3590
3580 SN$=LEFT$(B$(NC),CC-1)
3590 IF CC=LC(NC) THEN DE$="":GOTO3610
3600 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3610 B$(NC)=SN$+A$+DE$
3620 RETURN
3625 REM
3630 REM ** COMANDOS **
3640 REM
3680 IF FS>5 THEN 3710
3700 RETURN
3710 IF FS=9 THEN GOSUB 13000:RETURN
3720 IF FS=8 THEN GOSUB 4080:RETURN
3725 IF FS<8 THEN RETURN
3730 ON FS-9 GOSUB 3740,3910
3735 RETURN
3740 REM
3745 REM ** INSERCION **
3750 REM
3770 IF CC=LC(NC)THEN RETURN
3780 IF CC=1 THEN SN$="":GOTO3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DE$=RIGHT$(B$(NC),N)
3820 C$=BL$
3830 A$=SN$+C$+DE$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 REM
3890 PRINT B$(NC):S=-1
3900 IF RIGHT$(B$(NC),1)=PD$ THEN 3780
3905 RETURN
3909 REM
3910 REM ***** BORRADO *****
3920 REM
3930 IF TP(NC)<>1 THEN 3950
3940 IF MID$(B$(NC),2,1)=PD$ AND CC=1 THEN RETURN
3950 IF CC<1 THEN RETURN
3960 IF CC=LC(NC) THEN DE$="":GOTO 3980
3970 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3980 SN$=LEFT$(B$(NC),CC-1)
3990 IF TP(NC)=1 OR TP(NC)=2 THEN C$=NS$:GOTO 4010
4000 C$=AS$
4010 B$(NC)=SN$+DE$+C$
4020 XX=X:YY=Y
4030 X=CX(NC):Y=CY(NC):GOSUB 11000
4040 X=XX:Y=YY
4050 PRINT B$(NC):S=1:RETURN
4080 REM ******* ALINEADOS *******
4084 IF LC(NC)=1 THEN RETURN
4085 N1=LC(NC)-CC+1
4087 IF CC=1 THEN B$(NC)="":GOTO4100
4090 B$(NC)=LEFT$(B$(NC),CC-1)
4100 IF TP(NC)=1 OR TP(NC)=2 THEN 4150
4110 FOR I=1 TO N1
4120 B$(NC)=B$(NC)+BL$
4130 NEXT I
4140 GOTO  4190
4150 FOR I=1 TO N1
4160 B$(NC)=CHR$(32)+B$(NC)
4170 NEXT I
4180 IF RIGHT$(B$(NC),1)=PD$ THEN B$(NC)=BL$+LEFT$(B$(NC),LC(NC)-1)
4190 X=CX(NC):Y=CY(NC):GOSUB 11000
4210 PRINT B$(NC):RETURN
4270 REM CTRL PUNTO
4290 SWP=0
4300 IF CC=1 OR CC=LC(NC) THEN SWP=1:RETURN
4310 FOR I=1 TO LC(NC)
4320 IF MID$(B$(NC),I,1)=BL$ THEN I=LC(NC):SWP=1:RETURN
4330 NEXT I:RETURN
4340 REM
6000 REM
6001 REM *** GESTION DISCO ***
6010 REM -----------------------
6011 REM APERTURA CANAL COMANDOS
6012 REM -----------------------
6030 OPEN1,8,15
6035 RETURN
6200 REM --------------
6204 REM APERTURA FICHERO
6206 REM --------------
6210 OPEN2,8,3,NF$
6215 REM
6250 RETURN
6400 REM -----------------------
6402 REM CIERRE CANAL COMANDOS
6403 REM -----------------------
6410 CLOSE1
6415 RETURN
6490 REM
6600 REM -----------
6602 REM CIERRA FICHERO
6605 REM -----------
6610 CLOSE2
6620 RETURN
6800 REM ---------------
6801 REM APUNTA AL REGISTRO
6802 REM ---------------
6805 R1=PA:R2=0
6810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6815 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6820 RETURN
6825 REM
6900 REM CONTROLES SOBRE LOS FICHEROS
6905 GOSUB 6000:GOSUB 6200
6910 INPUT#1,A,B$,C,D
6915 GOSUB 6400:GOSUB 6600
6930 RETURN
7000 REM
7001 REM *** CARGA LOS DATA ***
7002 IF OPZ=5 THEN 7006
7003 L=LEN(VO$(OPZ))-3:L$=RIGHT$(VO$(OPZ),L):IF OPZ=0 THEN 7005
7004 U$="XOZ "+STR$(OPZ)+" "
7005 P=(30-L)/2:P$="":FORK=1TOP:P$=P$+" ":NEXT K
7006 PRINT"";:PRINTCHR$(14)
7008 PRINT"              .I.♥.    DICIONES _ORUM, ♥.♠."
7011 PRINT"   ";U$;"                          "
7012 PRINT"             ";P$;
7013 IF OPZ=0 THEN PRINT"  ";
7015 PRINT""L$;"";P$
7020 PRINT"                      "
7030 X=1:Y=22:GOSUB11000:PRINT"──────────────────────────────"
7033 X=25:Y=23:GOSUB11000:PRINT"─┴─┼─ : ♠/─♠"
7034 REM
7035 IF OPZ=0 THEN RETURN
7037 REM
7040 ON OZ GOTO 7042,7044,7046,7110,7110,7110
7042 LV=54:GOSUB20000:NUC=6:DS=11:GOTO7090
7044 LV=117:GOSUB20000:NUC=4:DS=2:GOTO7090
7046 LV=143:GOSUB20000:NUC=4:DS=13:GOTO7090
7090 FOR I=1 TO DS:READ D$(I),DX(I),DY(I):NEXTI
7095 FOR I=1 TO NUC:READ TP(I),LC(I),ND(I),CX(I),CY(I)
7097 REM
7100 NEXT I
7110 RETURN
8000 REM ***** GESTION DISCO *****
8005 IF I0=2 THEN 8060
8010 RE=1+RE
8015 IF RE>VAL(TD$) THEN RETURN
8020 PA=RE:GOSUB6800:INPUT#2,A$
8025 K1=1:I=0
8030 I=I+1:IF I=11 THEN RETURN
8035 K2=VAL(B2$((I-1)*6+3))
8040 IF K2=0 OR SK$(I)<>"S" THEN 8050
8045 B$(I)=MID$(A$,K1,K2)
8050 K1=K1+K2
8055 GOTO 8030
```

```
8060 A$="":I=0
8065 I=I+1:IF I=11 THEN 8085
8070 IF VAL(B2$((I-1)*6+3))=0 OR SK$(I)<>"S" THEN 8080
8075 A$=A$+B$(I)
8080 GOTO 8065
8085 REM ESCRIBE EL REGISTRO FORMADO
8090 R8=VAL(TD$)+1
8095 PA=R8:GOSUB6800:PRINT#2,A$
8100 TD$=STR$(R8)
8105 PA=1:GOSUB6800:PRINT#2,TD$
8110 RETURN
8112 REM ---------------------------------
9000 REM
9001 REM RUTINAS CALCULOS
9002 REM
9150 REM **** ERRORES ****
9152 PRINTHO$
9154 ON ERR GOTO 9160,9170
9160 X=2:Y=10:GOSUB11000:PRINT"■LA LONGITUD DEFINIDA PARA EL REGISTRO"
9164 Y=Y+2:GOSUB11000:PRINT"SUPERA EL MAXIMO PERMITIDO "
9166 GOTO 9500
9170 X=2:Y=10:GOSUB11000:PRINT"■┌─┌─ EN DISCO  "
9174 Y=Y+2:GOSUB11000:PRINT" ⁄⁄╮¯♠¯ FUERA DE LINEA "
9176 GOTO 9500
9500 REM RETORNO
9505 X=6:Y=20:GOSUB11000:PRINT"‼PULSE UNA TECLA PARA VOLVER"
9510 POKE198,0:WAIT 198,1:POKE198,0:RETURN
9800 REM *** ESPERA DE 1 TECLA ***
9805 X=4:Y=24:GOSUB11000:PRINT"■PULSAR UNA TECLA PARA CONTINUAR";
9810 POKE 198,0:WAIT 198,1
9815 POKE 198,0
9820 RETURN
9850 REM *** PETICION IMPRESION V,C ***
9855 X=3:Y=10:GOSUB11000:PRINT"‼INTRODUCIR OPCION :"
9860 X=7:Y=13:GOSUB11000:PRINT"⌧<≡1⌦‼ EN VIDEO"
9865 Y=Y+2:GOSUB11000:PRINT"⌧<≡2⌦‼ EN IMPRESORA"
9870 X=16:Y=Y+3:GOSUB11000:PRINT"■ELIJA ":X=22
9875 GET R$:IF R$<>"" THEN 9995
9880 GOSUB11000:PRINT"■?":FORT=1TO90:NEXTT
9885 GOSUB11000:PRINT" ":FORT=1TO90:NEXTT
9890 GOTO 9875
9995 R=VAL(R$):IF R<1 OR R>2 THEN 9875
9900 RETURN
10000 REM ************ DATA ************
10005 DATA 145,17,29,157,140,133,136,13,25,134,135,0,0,0,0
10007 DATA"◣    < ─┐ ⁄─⁄ >      ‼┐♠─♠ ─┌⁄╲⁄♠─"
10010 DATA"◣    ─♠─ ─╲│ ││  :  ‼ ¯♥└♠♠ ♠ ¯─¯─│♠"
10012 DATA"◣    ─♠─ └─┘     :  ‼ ¯♥└♠♠ ♠ ╲♠♠ ⌒─¯♠"
10013 DATA"◣    ─♠─  ⁄┐     :  ‼ ¯♥└♠♠ ♠──╲│♠     "
10015 DATA"◣    ─♠─  ┐O⁄    :  ‼ ¯♥└♠♠ ♠│♠ ┌  "
10017 DATA"◣          ⁄─1⁄  :  ‼ ♠⁄⁄└♠ ╲⁄│ ─┌¯ ⌒─╲┌⁄"
10019 DATA"◣          ⁄─3⁄  :  ‼ ╲⁄♥¯─│♠ ¯♠─♠─│¯─  "
10021 DATA"◣          ⁄─5⁄  :  ‼ │┌──♠ ¯♠─♠─│¯─  "
10023 DATA"◣          ⁄─7⁄  :  ‼ X⁄┐─│♠ ♠└ ╲¯⁄⁄  "
10025 DATA"◣          ⁄─8⁄  :  ‼ ╲⁄│ ─┌¯ ⌒─╲┌⁄   "
10027 DATA"◣    ─┐ ⁄─⁄      :  ‼ ♠└╲⁄¯♠ │ ─┌⁄X♠└╲¯♠"
10029 DATA"◣    ─│─┴+│      :  ‼ ┐╲¯ └♠ ♠│⁄♠    "
10030 REM ******* OPERADORES ********
10040 DATA"+","-","*","/","↑",">  ","<  ",">=","<=","= ","<>"
10050 REM ******* INSTRUCCIONES ********
10055 DATA"LEE",10,"ESC",10,"BOR",0,"BUS",3,"CAL",4,"DEC",4,"REP",10,"FIN",0
10113 REM ******** OPZ=1 ***************
10115 DATA"‾ESCRIPCION │IPO LON  ‾EC  ‾OL LIN",4,7
10116 DATA"1: ",4,9,"2: ",4,10,"3: ",4,11,"4: ",4,12,"5: ",4,13,"6: ",4,14
10117 DATA"7: ",4,15,"8: ",4,16,"9: ",4,17,"10: ",3,18
10118 DATA3,10,0,7,9
10119 DATA1,1,0,18,9
10120 DATA1,2,0,22,9
10121 DATA1,2,0,27,9
10122 DATA1,2,0,32,9
10123 DATA1,2,0,37,9
10130 REM ********* OZ=2 ***************
10131 DATA"⁄⌒╲│─‾ ─╲─│‾─┘‾",1,9
10132 DATA"└┌⁄│╲│ ⁄‾",24,9
10133 DATA3,6,0,16,9
10134 DATA1,2,0,35,9
10135 DATA3,1,0,15,15
10136 DATA3,1,0,33,15
10140 REM ********* OZ=3 ***************
10141 DATA"⁄UM.",3,8,"╲NSTRUCCION",11,8,"‾ARAMETRO",27,8
10142 DATA"1:",5,14,"2:",5,15,"3:",5,16,"4:",5,17,"5:",5,18
10143 DATA"6:",23,14,"7:",23,15,"8:",23,16,"9:",23,17,"10:",22,18
10144 DATA1,2,0,4,10
10145 DATA3,10,0,11,10
10146 DATA3,6,0,28,10
10147 DATA3,10,0,8,14
10200 REM *****************************
11000 REM
11001 REM **** POSICIONADO EN X,Y ****
11005 POKE 211,X
11010 POKE 214,Y
11015 SYS58732
11020 RETURN
12000 REM *** FASE 1, DEFINICIONES ***
12005 PRINTHO$:OZ=1:GOSUB 7000
12008 REM *** PREPARA MASCARA ***
12010 FOR J=2 TO 10
12012 K1=1+6*(J-1):K2=K1+5:P=0:IN=CY(K1-6)
12014 FOR K=K1 TO K2
12015 P=P+1
12016 TP(K)=TP(P):LC(K)=LC(P):ND(K)=ND(P):CX(K)=CX(P):CY(K)=IN+1
12017 NEXT K
12018 NEXT J
12019 IFG9<>0THENFORJ=1TO60:B$(J)=B2$(J):TP(J)=5:NEXTJ
12020 NUC=60:GOSUB 2000
12021 IF G9=0 THEN 12024
12022 K9=2
12023 FOR K=K9 TO K9+4:TP(K)=1:NEXT K:IF K9<56 THEN K9=K9+6:GOTO12023
12024 J=1
12025 GOSUB 3040
12027 IF FS=7 THEN RETURN
12030 IF FS=6 THEN 12020
12032 IF FS=9 THEN 12025
12040 G9=1
12042 REM ** TRANSFIERE LOS BUFFERS A B2$(*) Y PONE A CERO **
12045 FOR I=1 TO 60
12047 B2$(I)=B$(I):B$(I)=""
12050 NEXT I
12052 FOR I=1 TO 60STEP6
12054 I$=LEFT$(B2$(I),1)
12056 IF((I$<>CHR$(32)ORI$<>CHR$(35))ANDI$<>"")ANDB2$(I+1)<>"1"THENB2$(I+1)="3"
12058 V1=VAL(B2$(I+1)):V2=VAL(B2$(I+2)):V3=VAL(B2$(I+3))
12060 IF V1=1 AND V2<=V3 THEN B2$(I+2)=STR$(V3+1)
12062 NEXT I
12064 PRINTHO$:OZ=2:GOSUB 7000
12066 IN=CY(3)
12068 FOR I=5 TO 12STEP2
12070 TP(I)=TP(3):LC(I)=LC(3):ND(I)=ND(3):CX(I)=CX(3):IN=IN+1:CY(I)=IN
12077 TP(I+1)=TP(3):LC(I+1)=LC(3):ND(I+1)=ND(3):CX(I+1)=CX(4):CY(I+1)=IN
12078 NEXT I
12080 JJ=1
12082 FOR I=1 TO 5:Y=14+I
12084 FOR J=0TO1:X=2+18*J
12085 GOSUB11000:IFRIGHT$(B2$(JJ),1)="#"THENJJ=JJ+6:GOTO12087
12086 PRINT"≡";B2$(JJ):JJ=JJ+6
12087 NEXT J,I:JJ=0
12088 X=0:Y=12:GOSUB11000:PRINT"‼╲NTRODUCIR ⁄≡♥‼⁄ PARA CAMPOS A MEMORIZAR"
12090 X=0:Y=13:GOSUB11000:PRINT"■────────────────────────────────────────";
12091 NUC=12
12092 GOSUB 2000
12094 J=1
12096 GOSUB 3040
12098 IF FS=7 THEN RETURN
12100 IF FS=6 THEN 12092
12105 IF FS=9 THEN 12096
12107 IF B$(1)="######"THEN 12091
12110 PRINTHO$
12112 X=4:Y=8:GOSUB11000:PRINT"‼♥I LA UNIDAD DE DISCO SE VA":Y=Y+2
12114 GOSUB11000:PRINT"A EMPLEAR PULSAR LA TECLA ⁄≡♥‼⁄"
12116 X=X+4:Y=Y+3:GOSUB11000:PRINT"♥I NO, LA TECLA ⁄≡⁄‼⁄"
12118 GET R$:IF R$="" THEN 12118
12120 IF R$<>"N" AND R$<>"S" THEN 12118
12122 IF R$="N" THEN RETURN
12125 LR=0:FORI=3TO12
12127 IFB$(I)="S"ORB$(I)="♥"THENLR=LR+VAL(B2$((I-3)*6+3)):SK$(I-2)="S":GOTO12135
12130 SK$(I-2)=""
12135 NEXT I
12138 IF LR>90 THEN ERR=1:GOSUB9150:RETURN
```

```
12140 NF$=B$(1):GOSUB 6900
12142 IF A=62 THEN 12150
12143 IF A=74 OR A=70 THEN ERR=2:GOSUB 9150:A=0:GOTO12064
12144 GOSUB 6000:GOSUB 6200
12145 PA=1:GOSUB 6800:INPUT#2,TD$
12146 GOSUB 6600:GOSUB 6400
12148 RETURN
12150 REM ABRE EL FICHERO Y ESCRIBE REG.#1
12152 GOSUB 6000
12154 OPEN2,8,3,"0:"+NF$+",L,"+CHR$(LR+2)
12156 TD$="01":FORK=1TOLR-2:TD$=TD$+"#":NEXTK
12158 PA=1:GOSUB 6900
12160 PRINT#2,TD$
12162 CLOSE2
12164 GOSUB 6400
12170 RETURN
13000 REM *** AYUDA ***
13005 CO=X:CV=Y
13010 X=1:Y=23
13015 GOSUB11000:PRINTH$(1);"       ";
13020 FOR HE=2 TO 12
13025 GETR$:IFR$=""ORR$<>CHR$(13)THEN13025
13030 GOSUB11000:PRINTH$(HE)
13035 NEXT HE
13037 GETR$:IFR$=""ORR$<>CHR$(13)THEN13037
13040 GOSUB11000:PRINT"                    [illegible]";
13050 X=CO:Y=CV
13070 RETURN
14000 REM ***** EJECUCION PROGRAMA *****
14001 DE=0
14002 RE=1:IF VAL(TD$)>0 THEN GOSUB 6000:GOSUB 6200
14003 GOSUB26000
14004 NI=0
14006 NI=NI+1:IF NI=M3 THEN 14020
14008 IST$=IST$(NI):PER$=PER$(NI):NT=NT(NI)
14010 FORI=1TO10:P7$(I)=PAR$(NI,I):NEXTI
14011 I=0
14012 I=I+1:IF LEFT$(IST$,3)=FR$(I) THENI0=I:GOTO14016
14014 IF I<8 THEN 14012
14015 GOTO 14020
14016 ON I GOSUB 14100,14200,14300,14400,14600,14700,14800,14900
14017 IF ER<>0 THEN RETURN
14018 IF FI=1 THEN CLOSE1:CLOSE2:GOTO14020
14019 GOTO 14006
14020 X=2:Y=21:GOSUB11000:PRINT"█PULSAR UNA TECLA PARA VOLVER AL MENU"
14022 POKE198,0:WAIT198,1:POKE198,0:RETURN
14100 REM **** LEE ****
14105 OZ=6:NUC=10
14110 GOSUB 16000
14115 IF PER$="D" THEN GOSUB8000:RETURN
14120 IF PER$<>"V" THEN RETURN
14121 PRINTHO$
14122 GOSUB 2000
14124 GOSUB 3000
14126 IF FS=9 THEN 14124
14128 IF FS=6 THEN 14122
14130 IF FS=7 THEN RETURN
14132 RETURN
14133 REM --------------------------------
14200 REM **** ESCRIBE ****
14205 OZ=6:NUC=10
14210 GOSUB 16000
14215 IF PER$="D" THEN GOSUB 8000:RETURN
14220 FOR I=1 TO10:IF TP(I)<>0 THEN TP(I)=4
14225 NEXT I
14230 GOSUB 2000
14235 X=4:Y=21:GOSUB11000:PRINT"█PULSAR UNA TECLA PARA SEGUIR"
14240 POKE198,0:WAIT198,1:POKE198,1
14245 RETURN
14250 REM --------------------------------
14300 REM ***** BORRA *****
14305 IF PER$="V" THEN PRINTHO$:RETURN
14310 AA$="":FOR K=1 TO LR:AA$=AA$+"#":NEXTK
14315 FOR I=2 TO VAL(TD$):PA=I:GOSUB6800:PRINT#2,AA$:NEXT I
14320 TD$="01"+RIGHT$(AA$,LR-2)
14325 PA=1:GOSUB6800:PRINT#2,TD$
14330 RETURN
14335 REM --------------------------------
14400 REM **** BUSCA *****
14405 K=1:GOSUB 16100:P2=VAL(B2$((J-1)*6+3))
14410 IF P7$(2)="" THEN OP=10:GOTO14420
14412 GOTO 14440
14420 K9=ASC(B$(J))
14425 IF K9>47ANDK9<58 OR K9=32 THEN Z=VAL(B$(J)):CP$="":GOTO 14435
14430 CP$=B$(J)
14435 GOTO 14465
14440 FOR OP=6 TO11:IFLEFT$(P7$(2),2)=OP$(OP)THEN14450
14445 NEXT OP:ER=4:GOTO 25000
14450 K8=ASC(P7$(3))
14455 IF K8>47ANDK8<58 THEN Z=VAL(P7$(3)):CP$="":GOTO 14465
14460 CP$=P7$(3)
14465 IF J=1 THEN P1=1:GOTO 14485
14470 P1=1:FORI=1TOJ-1:IFSK$(I)<>"S"THEN14480
14475 P1=P1+VAL(B2$((I-1)*6+3))
14480 NEXT I
14485 FOR II=2 TO VAL(TD$)
14490 PA=II:GOSUB6800:INPUT#2,A$
14495 GOSUB 15010:IFFC=1THEN14515
14500 NEXTII:PRINTHO$:X=10:Y=10:GOSUB11000:PRINT"▮NO HALLADO ";P7$(1)
14505 FOR I=1 TO 10:IFSK$(I)="S"THENB$(I)=""
14510 NEXT I:GOTO14525
14515 GOSUB 8025
14520 PRINTHO$:X=14:Y=10:GOSUB11000:PRINT"▮HALLADO ";P7$(1)
14525 FORI=1TO1000:NEXTI:RETURN
14530 REM --------------------------------
14600 REM ***** CALCULA *****
14605 IFASC(P7$(2))<48ORASC(P7$(2))>57THEN K=2:GOSUB16100:Z1=VAL(B$(J)):GOTO1461
5
14610 Z1=VAL(P7$(2))
14615 IFASC(P7$(4))<48ORASC(P7$(4))>57THEN K=4:GOSUB16100:Z2=VAL(B$(J)):GOTO1462
5
14620 Z2=VAL(P7$(4))
14625 I=0
14630 I=I+1:IF I=6 THEN ER=5:GOTO25000
14635 IFLEFT$(P7$(3),1)=OP$(I)THEN14645
14640 GOTO 14630
14645 ON I GOTO 14655,14660,14665,14670,14675
14650 K=1:GOSUB16100:B$(J)=MID$(STR$(Z3),2,LEN(STR$(Z3))-1):LC=VAL(B2$((J-1)*6+3
))
14651 LE=LEN(B$(J)):IF LE>LC THEN B$(J)=LEFT$(B$(J),LC)
14652 IF LE=LC THEN 14654
14653 FOR OT=1 TO LC-LE:B$(J)="0"+B$(J):NEXT OT
14654 RETURN
14655 Z3=Z1+Z2:GOTO14650
14660 Z3=Z1-Z2:GOTO14650
14665 Z3=Z1*Z2:GOTO14650
14670 Z3=Z1/Z2:GOTO14650
14675 Z3=Z1↑Z2:GOTO14650
14680 REM --------------------------------
14700 REM ***** DECIDE ******
14705 K=1:GOSUB16100:A$=B$(J)
14710 IFASC(P7$(3))>47ANDASC(P7$(3))<58THENZ=VAL(P7$(3)):CP$="":GOTO14760
14715 IFLEFT$(P7$(3),1)=CHR$(39)THENP1=2:I=1:GOTO14725
14720 GOTO14740
14725 I=I+1:IF I=11THENP2=9:GOTO14750
14730 IFMID$(P7$(3),I,1)=CHR$(39)THENP2=I-2:GOTO14750
14735 GOTO 14725
14740 K=3:GOSUB16100:CP$=B$(J)
14750 CP$=MID$(P7$(3),P1,P2)
14752 OP=5
14755 OP=OP+1:IFOP=12THENER=4:GOTO25000
14760 IF LEFT$(P7$(2),2)=OP$(OP)THEN 14762
14761 GOTO 14755
14762 GOSUB15000:IF FC=0 THEN RETURN
14764 I=0
14766 I=I+1:IF I=M3+1 THEN P5$=P7$(4):GOTO14770
14768 IFVAL(P7$(4))=NT(I)THEN14774
14769 GOTO 14766
14770 PRINTHO$:PRINT"▮▮▮▮QQQQQQQQQQLA INSTRUCCION ";P5$;"NO EXISTE"
14772 PRINT"QQQQ▮PULSAR UNA TECLA PARA VOLVER AL MENU":POKE198,0:WAIT198,1:ER=1
RETURN
14774 IST$=IST$(I):PER$=PER$(I):NT=NT(I):K=0
14776 K=K+1:IF K=11 THEN I=0:GOTO14780
14778 P7$(K)=PAR$(I,K):GOTO 14776
14780 I=I+1:IF I=9 THEN ER=1:GOTO25000
14782 IFLEFT$(IST$,3)=FR$(I)THEN14785
14784 GOTO 14780
14785 DE=1
```

```
14786 ON I GOSUB14100,14200,14300,14400,14600,14700,14800,14900
14790 RETURN
14792 REM --------------------------------
14800 REM ***** REPITE ******
14805 IFASC(PER$)>47ANDASC(PER$)<58THENN=VAL(PER$):GOTO14815
14810 K=0:P7$(K)=PER$:GOSUB16100:N=VAL(B$(J))
14815 FOR WI=1 TO 10:PP$(WI)=P7$(WI):NEXTWI
14820 FOR WI=1 TO N
14825 FOR WJ=1 TO 10:IF LEFT$(PP$(WJ),3)="FIN" THEN 14890
14830 FOR WK=1 TO M3:IFVAL(PP$(WJ))=NT(WK)THEN14840
14835 NEXTWK:P5$=PP$(WJ):GOTO 14770
14840 IST$=IST$(WK):PER$=PER$(WK):NT=NT(WK)
14845 IF LEFT$(IST$,3)<>"RIP" THEN 14860
14850 PRINTHO$:X=1:Y=12:GOSUB11000:PRINT"▮NO SE PUEDE REPETIR LA INSTRUCCION";NT
14855 Y=Y+3:GOSUB11000:PRINT"PULSE UNA TECLA":POKE198,0:WAIT198,1:ER=5:RETURN
14860 FORWW=1TO10:P7$(WW)=PAR$(WK,WW):NEXTWW
14865 FOR IK=1 TO8:IF LEFT$(IST$,3)=FR$(IK) THEN 14875
14870 NEXT IK:ER=1:GOTO 25000
14875 I0=IK
14880 ON I0 GOSUB14100,14200,14300,14400,14600,14700,14800,14900
14885 NEXT WJ
14890 NEXT WI
14895 RETURN
14897 REM ----------------------------------
14900 REM **** FIN ****
14902 IF DE=1 THEN 14910
14905 CLOSE1:CLOSE2
14910 X=12:Y=19:GOSUB11000:PRINT"▮FIN  PROGRAMA":FI=1:RETURN
14920 REM --------------------------------
15000 REM ****** COMPARACION ******
15005 P1=1:P2=10
15010 A1$=MID$(A$,P1,P2)
15015 K7=VAL(A1$)
15020 IFCP$=""THENZ1=K7:FZ=0:GOTO15050
15025 FZ=1
15030 IFLEN(A1$)=>10THENA1$=LEFT$(A1$,10):GOTO 15040
15032 K5=10-LEN(A1$)
15035 FOR Q=1 TO K5:A1$=A1$+CHR$(32):NEXTQ
15040 IF LEN(CP$)=>10THEN CP$=LEFT$(CP$,10):GOTO 15050
15042 K5=10-LEN(CP$)
15045 FOR Q=1 TO K5:CP$=CP$+CHR$(32):NEXTQ
15050 O=OP-5+(FZ*6)
15051 IF O>6 THEN 15055
15052 ON O GOTO 15060,15080,15100,15120,15140,15160
15055 ON O-6 GOTO 15070,15090,15110,15130,15150,15170
15060 IF Z1>Z THEN FC=1:RETURN
15062 FC=0:RETURN
15070 IF A1$>CP$ THEN FC=1:RETURN
15072 FC=0:RETURN
15080 IF Z1<Z THEN FC=1:RETURN
15082 FC=0:RETURN
15090 IF A1$<CP$ THEN FC=1:RETURN
15092 FC=0:RETURN
15100 IF Z1>=Z THEN FC=1:RETURN
15102 FC=0:RETURN
15110 IF A1$>=CP$ THEN FC=1:RETURN
15112 FC=0:RETURN
15120 IF Z1<=Z THEN FC=1:RETURN
15122 FC=0:RETURN
15130 IF A1$<=CP$ THEN FC=1:RETURN
15132 FC=0:RETURN
15140 IF Z1=Z THEN FC=1:RETURN
15142 FC=0:RETURN
15150 IF A1$=CP$ THEN FC=1:RETURN
15152 FC=0:RETURN
15160 IF Z1<>Z THEN FC=1:RETURN
15162 FC=0:RETURN
15170 IF A1$<>CP$ THEN FC=1:RETURN
15172 FC=0:RETURN
16000 REM **** BUSCA CAMPO ****
16005 J=0:I=-5
16010 I=I+6:IF I>55 THEN 16045
16015 J=J+1
16016 K=0
16020 K=K+1:IF K=11 THEN TP(J)=0:GOTO 16040
16025 IF P7$(K)=B2$(I) THEN 16035
16030 GOTO 16020
16035 D$(J)=B2$(I):TP(J)=VAL(B2$(I+1)):LC(J)=VAL(B2$(I+2)):ND(J)=VAL(B2$(I+3))
16037 DX(J)=VAL(B2$(I+4)):CY(J)=VAL(B2$(I+5)):CX(J)=(DX(J)+11):DY(J)=CY(J)
16040 GOTO 16010
16045 RETURN
16050 REM --------------------------------
16100 REM **** BUSCA CAMPO FC=1 ****
16105 J=0:I=-5
16110 I=I+6:IF I>55 THEN ER=3:GOTO25000
16115 J=J+1
16120 IF P7$(K)=B2$(I) THEN 16130
16125 GOTO 16110
16130 RETURN
16135 REM --------------------------------
20000 REM RESTORE DATA
20005 RESTORE
20010 FOR I=0 TO 62:READ D:NEXTI
20020 FOR I=1 TO NV:READ X$:NEXTI
20025 FORK=1 TO 3
20030 FOR I=0 TO 62:READ D:NEXTI
20035 NEXT K
20045 IF LV<=0 THEN RETURN
20050 FOR I=1 TO LV:READ LV$:NEXTI
20100 RETURN
25000 REM ****** ERROR *******
25010 PRINTHO$:X=6:Y=9:GOSUB11000:PRINT"▮┌──┌─ ";ER;" PULSAR UNA TECLA"
25015 POKE198,0:WAIT198,1:POKE198,0:RETURN
26000 REM PUESTAS A CERO
26010 FORK9=1TO13
26015 TP(K9)=4:LC(K9)=1:ND(K9)=0:D$(K9)="":B$(K9)=""
26020 CX(K9)=0:CY(K9)=0:DX(K9)=0:DY(K9)=0
26025 NEXT K9
26030 RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - COMMODORE 64

B$(*) = matriz de los buffers (1 por campo de introducción)
LC(*) = longitud del campo de introducción
ND(*) = número de decimales en el campo de introducción
CX(*), **CY(*)** = coordenadas en el vídeo del campo de introducción
DX(*), **DY(*)** = coordenadas en el vídeo de la descripción
TF(*) = matriz numérica con los códigos ASCII de las teclas de comando
D$(*) = descripciones en las máscaras video
VO$(*) = voces del menú principal
BL$ = blank
PD$ = punto decimal
AS$ = carácter "–"; identifica los campos de introducción alfanuméricos
NS$ = carácter "#"; identifica los campos de introducción numéricos
HO$ = borrado de pantalla
NUC = número de campos de introducción en la máscara video activa
DS = número de descripciones en la máscara video
B2$(*) = matriz auxiliar para las descripciones
IST$(*) = matriz para instrucciones
PER$(*) = cadenas de identificación para periféricos
PAR$ (*,*) = parámetros en las instrucciones
G5 = flag de llamada al bucle de introducción de instrucciones
M3 = número máximo de instrucciones
OP$(*) = operadores
NP(*) = número parámetros
FR$(*) = matriz para el reconocimiento instrucciones
NF$ = nombre fichero datos
PA = registro actual
TD$ = último registro introducido, en forma alfanumérica
I0 = índice para gestión del disco
FC = flag de comparación
ER = indicador de error



```
10    REM  ---------------------------
20    REM  ORDENADOR PARA EL ORDENADOR
25    REM        VERSION DOS
30    REM  ---------------------------
50    MP$ = "EL ORDENADOR PARA EL ORDEN
      ADOR":
      DIM B$(60),TP(60),ND(60),LC(60),D
      $(60),DX(60),DY(60),CX(60),CY(60)
      ,IS$(99),PA$(99,10),PE$(99):
      ONERR  GOTO 40000
100   GOSUB 1300:
      IF OP = NV THEN  TEXT :
      HOME :
      END
115   FOR I = 1 TO 7:
         READ FR$(I):
      NEXT I:
      FOR I = 1 TO 6:
         READ CF$(I):
      NEXT I:
      FOR I = 1 TO 4:
         READ OP$(I):
      NEXT I:
      FOR I = 1 TO 9:
         READ ER$(I):
      NEXT I:
      ON OP GOSUB 5000,8000,12000,14000

140   GOTO 100
500   REM
530   TEXT :
      HOME :
      BP$ =  CHR$ (7):
      D$ =  CHR$ (4):
      DR$ = "1":
      B$ = "          ":
      BA$ = B$ + B$ + B$ + B$:
      B$ = "__________":
      BU$ = B$ + B$ + B$ + B$:
      B$ = "==========":
      BE$ = B$ + B$ + B$ + B$
565   HE$ = "CTRL E. INTRODUCCION - CTR
      L R. VUELVE AL MENU - CTRL A,S,Z,
      W. MUEVEN EN LAS 4 DIRECCIONES -
      CTRL I. INSERCION - CTRL D. BORRA
       - ESC . ANULACION          PULSAR
      UNA TECLA":
      B$ = "":
      TEXT :
      HOME
580   INVERSE :
      PRINT BA$;:
      FOR I = 1 TO 21:
         PRINT " ";:
         HTAB 40:
         PRINT " ";:
      NEXT I:
      PRINT BA$;:
      NORMAL :
      VTAB 2:
      HTAB 4:
      PRINT "E. G. S.      EDICIONES FOR
      UM, S.A.":
      TI$ = "** " + MP$ + " **":
      YT = 4:
      GOSUB 2700:
      NORMAL :
      TI$ = "  " + MP$ + "  ":
      INVERSE :
      GOSUB 2700
660   NORMAL :
      RETURN
1000  REM
1010  X1 = X - 1:
      Y1 = Y - 1:
      CH =  SCRN( X1,2 * Y1) + 16
       *  SCRN( X1,2 * Y1 + 1):
      HTAB X:
      VTAB Y:
      FLASH :
      IF FC = 0 THEN  INVERSE
1040  PRINT  CHR$ (CH);:
      HTAB X:
      NORMAL :
      RETURN
1100  REM
1110  X = X1 + 1:
      Y = Y1 + 1:
      NORMAL :
      HTAB X:
      VTAB Y:
      PRINT  CHR$ (CH);:
      HTAB X:
      RETURN
1200  REM
1210  A0 =  PEEK ( - 16384):
      IF A0 > 128 THEN  POKE  - 16368,0
      :
      A0 = A0 - 128:
      RETURN
1220  GOTO 1210
1300  REM
1330  GOSUB 500:
      RESTORE :
      READ NV:
      IF NV = 0 THEN  RETURN
1350  L = 0:
      FOR I = 1 TO NV:
         READ VO$(I):
         LL =  LEN (VO$(I)):
         IF LL > L THEN L = LL
1370  NEXT I:
      XI = (40 - L) / 2:
      YI = ((24 - NV * 2) / 2) + 1:
      X = XI - 2:
      Y = YI:
      FOR I = 1 TO NV:
         HTAB X:
         VTAB Y:
         PRINT I;" - ";VO$(I):
         Y = Y + 2:
      NEXT I:
      Y = Y + 1
1410  HTAB X:
      VTAB Y:
      PRINT "CUAL ? ";:
      GOSUB 1200:
      OP$ =  CHR$ (A0):
      OP =  VAL (OP$):
      IF OP < 1 OR OP > NV THEN
       PRINT BP$;:
      GOTO 1410
1440  FLASH :
      PRINT OP:
      NORMAL :
      FOR I = 1 TO 1000:
      NEXT I:
      RETURN
2000  REM
2030  GOSUB 500:
      HTAB 3:
      VTAB 23:
      PRINT IN$;:
      IF CO <  > 0 THEN  PRINT " ";CO:
2040  FOR IC = 1 TO NC:
         HTAB DX(IC):
         VTAB DY(IC):
         PRINT D$(IC):
         B$(IC) =  LEFT$ (B$(IC),LC(IC)
         ):
         L =  LEN (B$(IC)):
         IF L < LC(IC) THEN B$(IC)
          = B$(IC) +  LEFT$ (BA$,LC(IC)
          - L)
2080     IF TP(IC) <  > 1 THEN 2120
2085     IF ND(IC) = 0 THEN 2112
2090     PD = ND(IC):
         C1 = LC(IC) - PD - 1:
         C2 = LC(IC) - C1 - 1:
         B$(IC) =  LEFT$ (BA$,C1)
          + "." +  LEFT$ (BA$,C2):
         A$ =  LEFT$ (BE$,C1) + "."
          +  LEFT$ (BE$,C2):
         GOTO 2160
2112     B$(IC) =  LEFT$ (BA$,LC(IC)):
         A$ =  LEFT$ (BE$,LC(IC)):
         GOTO 2160
2120     IF TP(IC) <  > 4 AND TP(IC)
          <  > 5 THEN B$(IC) =  LEFT$ (
         BA$,LC(IC)):
         Z$ = "":
         L = 0:
         GOTO 2150
2130     Z$ = B$(IC):
         GOSUB 2500:
         L =  LEN (Z$):
         B$(IC) = Z$ +  MID$ (BA$,1,LC(
         IC) - L)
2150     A$ = Z$ +  MID$ (BU$,1,LC(IC)
          - L)
2160     HTAB CX(IC):
         VTAB CY(IC):
         PRINT A$:
      NEXT IC:
      RETURN
2500  REM
2530  L =  LEN (Z$):
      IF L = 0 THEN  RETURN
2535  FOR I = 1 TO L:
         C =  ASC ( MID$ (Z$,I,1)):
         IF C <  > 32 THEN I1 = I:
         GOTO 2560
2550  NEXT I:
      Z$ = "":
      RETURN
2560  FOR I = L TO 1 STEP  - 1:
         C =  ASC ( MID$ (Z$,I,1)):
         IF C <  > 32 THEN I2 = I:
         GOTO 2590
2580  NEXT I
2590  KO = I2 - I1 + 1:
      Z$ =  MID$ (Z$,I1,KO):
      RETURN
2700  REM
2730  XT = (42 -  LEN (TI$)) / 2:
      HTAB XT:
      VTAB YT:
      PRINT TI$;:
      RETURN
2800  REM
2830  A$ =  LEFT$ (BA$,38):
      HTAB 2:
      VTAB YT:
```

```
      PRINT A$;:
      RETURN
2900  REM
2930  Z2$ = "":
      A =  LEN (Z$):
      FOR I = 1 TO A:
         Z1$ =  MID$ (Z$,I,1):
         IF Z1$ <  > " " THEN Z2$
          = Z2$ + Z1$
2950  NEXT I:
      Z$ = Z2$:
      RETURN
3000  REM
3030  CC = 1:
      FC = 1:
      HTAB 27:
      VTAB 23:
      PRINT "AYUDA.CTRL-Y"
3040  XC = CX(CC):
      YC = CY(CC):
      LO = LC(CC):
      GOSUB 3500:
      IF AO = 5 OR AO = 18 OR AO
       = 27 THEN  RETURN
3070  IF AO <  > 26 THEN 3100
3080  CC = CC + 1:
      IF CC > NC THEN CC = 1
3085  IF TP(CC) = 4 THEN 3080
3090  GOTO 3040
3100  IF AO <  > 23 THEN 3140
3110  CC = CC - 1:
      IF CC < 1 THEN CC = NC
3120  IF TP(CC) = 4 THEN 3110
3130  GOTO 3040
3140  IF TP(CC) <  > 1 THEN B$(CC)
       = C$:
      GOTO 3080
3150  PD = 0:
      FOR I = 1 TO LC(CC):
         C =  ASC ( MID$ (C$,I,1)):
         IF (C < 48 OR C > 57) AND C
          <  > 32 AND C <  > 46 AND C
          <  > 45 THEN B$(CC) = C$:
         PRINT BP$;:
         GOTO 3040
3180     IF C = 46 THEN PD = PD + 1
3190  NEXT I:
      IF PD > 1 THEN B$(CC) = C$:
      PRINT BP$;:
      GOTO 3040
3200  C$ =  MID$ (C$,1,IC):
      Z$ = C$:
      GOSUB 2500:
      C$ = Z$:
      KO = LC(CC) -  LEN (C$):
      A$ =  MID$ (BA$,1,KO):
      A$ = A$ + C$:
      B$(CC) = A$:
      A$ =  MID$ (BE$,1,KO):
      A$ = A$ + C$:
      HTAB XC:
      VTAB YC:
      PRINT A$:
      GOTO 3080
3500  REM
3530  IC = 0:
      C$ = B$(CC):
      IF LO = 0 THEN  RETURN
3535  IF CC = 0 THEN  HTAB XC:
      VTAB YC:
      PRINT  MID$ (BU$,1,LO)
3540  X = XC:
      Y = YC
3545  IF  MID$ (C$,IC + 1,1) = "."
       AND TP(CC) = 1 THEN IC = IC
       + 1:
      X = X + 1
3550  GOSUB 1000:
      GOSUB 1200:
      GOSUB 1100
3560  IF AO = 5 OR AO = 23 OR AO
       = 26 OR AO = 18 OR AO = 27
       THEN  RETURN
3565  IF AO = 9 THEN  GOSUB 3700:
      GOTO 3560
3567  IF AO = 4 THEN  GOSUB 3800:
      GOTO 3550
3570  IF AO = 25 THEN  GOSUB 13000:
      GOTO 3550
3580  IF AO <  > 13 THEN 3620
3582  IF IC = 0 THEN 3590
3585  IF TP(CC) = 1 AND  MID$ (C$,IC,1)
       = "." THEN IC = IC - 1
3590  C$ =  MID$ (C$,1,IC):
      A$ =  MID$ (BA$,1,LO - IC):
      C$ = C$ + A$:
      A$ =  MID$ (BU$,1,LO - IC):
      IF TP(CC) = 1 THEN A$ =  MID$ (BE
      $,1,LO - IC)
3605  HTAB XC + IC:
      PRINT A$:
      RETURN
3620  IF AO <  > 1 THEN 3650
3630  IF IC = 0 THEN  PRINT BP$;:
      GOTO 3550
3640  IC = IC - 1:
      X = X - 1:
      GOTO 3550
3650  IF IC = LO THEN  PRINT BP$;:
      GOTO 3550
3655  IF AO = 0 THEN 3550
3660  IF AO = 19 THEN IC = IC + 1:
      X = X + 1:
      GOTO 3545
3670  CH$ =  CHR$ (AO):
      PRINT CH$;:
      C$ =  MID$ (C$,1,IC) + CH$
       +  MID$ (C$,IC + 2,39):
      IC = IC + 1:
      X = X + 1:
      GOTO 3545
3700  REM
3722  IF IC = LO OR TP(CC) = 1 THEN AO
       = 0:
      RETURN
3725  Z$ = C$:
      GOSUB 2500:
      C$ = Z$
3730  FC = 0:
      GOSUB 1000:
      GOSUB 1200:
      GOSUB 1100:
      IF AO = 5 OR AO = 4 OR AO = 23
       OR AO = 26 OR AO = 18 OR AO
       = 27 OR AO = 25 OR AO = 1
       OR AO = 19 OR AO = 13 THEN FC
       = 1:
      RETURN
3750  CH$ =  CHR$ (AO):
      C$ =  MID$ (C$,1,IC) + CH$
       +  MID$ (C$,IC + 1,LO - IC
       - 1):
      L =  LEN (C$):
      A$ =  MID$ (BU$,1,LO - L):
      A$ = C$ + A$:
      HTAB XC:
      PRINT A$:
      X = X + 1:
      IC = IC + 1:
      IF IC < LO THEN 3730
3790  FC = 1:
      RETURN
3800  REM
3830  IF IC = LO OR TP(CC) = 1 THEN
       RETURN
3840  Z$ = C$:
      GOSUB 2500:
      C$ = Z$:
      C$ =  MID$ (C$,1,IC) +  MID$ (C$,
      IC + 2,LO - IC - 1):
      L =  LEN (C$):
      A$ =  MID$ (BU$,1,LO - L):
      A$ = C$ + A$:
      HTAB XC:
      PRINT A$:
      RETURN
5000  REM
5030  NC = 60:
      J = 1:
      FOR I = 1 TO 60 STEP 6:
         B$(I) = VN$(J):
         B$(I + 1) =  STR$ (VT(J)):
         B$(I + 2) =  STR$ (VL(J)):
         B$(I + 3) =  STR$ (VD(J))
5070     B$(I + 4) =  STR$ (VX(J)):
         B$(I + 5) =  STR$ (VY(J)):
         FOR K = I TO I + 5:
            TP(K) = 4:
            ND(K) = 0:
            D$(K) = "":
            DX(K) = 1:
            DY(K) = 1:
         NEXT K:
         LC(I) = 10:
         LC(I + 1) = 1:
         FOR K = I + 2 TO I + 5:
            LC(K) = 2:
         NEXT K:
         CX(I) = 4:
         CX(I + 1) = 17:
         CX(I + 2) = 21:
         CX(I + 3) = 26:
         CX(I + 4) = 31:
         CX(I + 5) = 36
5110     FOR K = I TO I + 5:
            CY(K) = J + 9:
         NEXT K:
         J = J + 1:
      NEXT I:
      D$(1) = "NOMB":
      D$(2) = "TIP":
      D$(3) = "LON":
      D$(4) = "DEC":
      D$(5) = "COL":
      D$(6) = "LIN":
      FOR K = 1 TO 6:
         DY(K) = 8:
      NEXT K:
      DX(1) = 4:
      DX(2) = 16:
      DX(3) = 21:
      DX(4) = 26:
      DX(5) = 31:
      DX(6) = 36
5160  IN$ = "DEFINICIONES FASE 1":
      CO = 0:
      GOSUB 2000:
      FOR I = 1 TO 60 STEP 6:
         TP(I) = 3:
         FOR K = I + 1 TO I + 5:
            TP(K) = 1:
         NEXT K:
      NEXT I:
      GOSUB 3000:
      IF AO = 18 THEN  RETURN
5210  IF AO = 27 THEN 5160
5220  J = 1:
      FOR IV = 1 TO 60 STEP 6:
         Z$ = B$(IV):
         GOSUB 2900:
         B$(IV) = Z$:
         VN$(J) = B$(IV):
         VT(J) =  VAL (B$(IV + 1)):
         VL(J) =  VAL (B$(IV + 2)):
         VD(J) =  VAL (B$(IV + 3)):
         VX(J) =  VAL (B$(IV + 4)):
         VY(J) =  VAL (B$(IV + 5)):
```

```
        J = J + 1:
     NEXT IV:
     FOR IV = 1 TO 10:
        IF VT(IV) <  > 1 THEN VT(IV)
         = 3
5266    IF VL(IV) <  = 0 THEN VL(IV)
         = 1
5267    IF VD(IV) < 0 THEN VD(IV)
         = 0
5268    IF VX(IV) < 2 THEN VX(IV)
         = 2
5269    IF VY(IV) < 5 THEN VY(IV)
         = 5
5270    IF VX(IV) > 39 THEN VX(IV)
         = 39
5271    IF VY(IV) > 23 THEN VY(IV)
         = 23
5272 NEXT IV:
     NC = 12:
     J = 16:
     D$(1) = "NOMB.FICHERO":
     D$(2) = "LONGITUD":
     FOR I = 3 TO 12:
        D$(I) = VN$(I - 2):
     NEXT I:
     TP(1) = 3:
     TP(2) = 1:
     LC(1) = 6:
     LC(2) = 2:
     DX(1) = 3:
     DY(1) = 8:
     DX(2) = 25:
     DY(2) = 8:
     CX(1) = 16:
     CX(2) = 35:
     CY(1) = 8:
     CY(2) = 8:
     FOR I = 3 TO 12 STEP 2:
        DX(I) = 7
5340    DY(I) = J:
        CX(I) = 18:
        CY(I) = J:
        J = J + 1:
     NEXT I:
     J = 16:
     FOR I = 4 TO 12 STEP 2:
        DX(I) = 23:
        DY(I) = J:
        CX(I) = 35:
        CY(I) = J:
        J = J + 1:
     NEXT I:
     FOR I = 3 TO 12:
        TP(I) = 1:
        LC(I) = 1:
     NEXT I:
     IN$ = "DEFINICIONES FASE 2"
5410 GOSUB 2000:
     HTAB 3:
     VTAB 14:
     PRINT "INTROD. '1' PARA CAMPOS A
     MEMORIZAR":
     GOSUB 3000:
     IF AO = 18 THEN  RETURN
5440 IF AO = 27 THEN 5410
5450 J = 1:
     VD = 0:
     FOR I = 1 TO 10:
        VM(I) =  VAL (B$(I + 2)):
        IF VM(I) = 1 THEN VI(I)
         = J:
        J = J + VL(I):
        VD = VD + 1
5467 NEXT I:
     DD = 0:
     IF B$(1) <  >  LEFT$ (BA$,6)
      THEN DD = 1
5480 MX =  VAL (B$(2)):
     NF$ = B$(1):
     LR = 0:
     FOR I = 1 TO 10:
        IF VM(I) = 1 THEN LR = LR
         + VL(I)
5510 NEXT I:
     IF DD = 0 OR LR = 0 THEN DD
      = 0:
     RETURN
5520 GOSUB 6000:
     NR = 1:
     :
     GOSUB 6200:
     MX =  VAL (AA$):
     GOSUB 6600:
     RETURN
5540 GOSUB 6600:
     GOSUB 6000:
     AA$ = "1":
     NR = 1:
     GOSUB 6400:
     GOSUB 6600:
     MX = 1:
     GOTO 140
6000 REM
6030 PRINT D$:
     PRINT D$;"OPEN";NF$ + ",D"
      + DR$;",L";LR + 2:
     PRINT D$:
     RETURN
6200 REM
6230 PRINT D$:
     PRINT D$;"READ";NF$;",R";NR:
     INPUT AA$:
     PRINT D$:
     RETURN
6400 REM
6430 PRINT D$:
     PRINT D$;"WRITE";NF$;",R";NR:
     PRINT  CHR$ (34) + AA$:
     PRINT D$:
     RETURN
6600 REM
6630 PRINT D$:
     PRINT D$;"CLOSE";NF$:
     PRINT D$:
     RETURN
8000 REM
8030 NC = 13:
     J = 14:
      D$(1) = "NUM.":
      D$(2) = "INSTRUCCION":
      D$(3) = "PARAM.":
      TP(1) = 1:
      TP(2) = 3:
      DX(1) = 7:
      CX(1) = 7:
      DX(2) = 14:
      CX(2) = 14:
      DX(3) = 29:
      CX(3) = 29:
      FOR I = 1 TO 3:
         CY(I) = 10:
         DY(I) = 8:
      NEXT I
8060  LC(1) = 2:
      LC(2) = 10:
      LC(3) = 6:
      TP(3) = 3:
      FOR I = 4 TO 13 STEP 2:
         LC(I) = 10:
         DX(I) = 1:
         DY(I) = 1:
         CX(I) = 8:
         CY(I) = J:
         D$(I) = "":
         TP(I) = 3:
         J = J + 1:
      NEXT I:
      J = 14:
     FOR I = 5 TO 13 STEP 2:
        LC(I) = 10:
        DX(I) = 1:
        DY(I) = 1:
        CX(I) = 24:
        CY(I) = J:
        D$(I) = "":
        TP(I) = 3
8140    J = J + 1:
     NEXT I
8150 IN$ = "INSTRUCCIONES":
     GOSUB 2000
8160 GOSUB 3000:
     IF AO = 18 THEN  RETURN
8170 IF AO = 27 THEN 8150
8175 IF  LEFT$ (B$(2),4) = "FIN"
      THEN  RETURN
8180 IS$ = B$(2):
     GOSUB 8500:
     IF ER <  > 0 THEN  GOSUB 30000:
     GOTO 8160
8195 NL =  VAL (B$(1)):
     IF NL <  = 0 THEN ER = 4:
     GOSUB 30000:
     GOTO 8160
8197 IS$(NL) = IS$:
     IF IS$ <  > "" THEN 8200
8198 FOR I = 1 TO 10:
        PA$(NL,I) = "":
     NEXT I:
     PE$(NL) = "":
     GOTO 8150
8200 PE$(NL) = "":
     FOR I = 1 TO 10:
        PA$(NL,I) = "":
     NEXT I:
     ON IS GOSUB 8600,8600,8700,8800,9
     000,9100,9300:
     IF ER = 0 THEN PE$(NL) = PE$:
     GOTO 8150
8220 GOSUB 30000:
     GOTO 8160
8500 REM
8530 Z$ = IS$:
     GOSUB 2900:
     Z$ =  LEFT$ (Z$,3):
     IS$ = Z2$:
     IF IS$ = "" THEN ER = 0:
     RETURN
8540 FOR I = 1 TO 7:
        IF FR$(I) = Z$ THEN ER = 0:
        IS = I:
        RETURN
8560 NEXT I:
     ER = 1:
     RETURN
8600 REM
8610 Z$ = B$(3):
     GOSUB 2900:
     Z$ =  LEFT$ (Z$,1):
     VK = 1:
     IF Z$ <  > "V" AND Z$ <  > "D"
      THEN ER = 3:
     RETURN
8625 IF Z$ = "D" AND DD = 0 THEN ER
      = 2:
     RETURN
8630 PE$ = Z$:
     FOR IV = 4 TO 13:
        Z$ = B$(IV):
        GOSUB 2900:
        VR$ = Z$:
        GOSUB 9500:
        IF ER <  > 0 THEN  RETURN
8655    IF PE$ = "D" AND VM(VR)
         <  > 1 AND VR$ <  > ""
         THEN ER = 6:
        RETURN
8660    IF VR$ <  > "" THEN PA$(NL,VK)
```

```
         = VR$:
      VK = VK + 1
8670  NEXT IV:
      ER = 0:
      IF VK = 1 THEN ER = 5
8680  RETURN
8700  REM
8710  Z$ = B$(3):
      GOSUB 2900:
      PE$ = Z$:
      ER = 0:
      IF  LEFT$ (PE$,1) <  > "V"
       AND  LEFT$ (PE$,1) <  > "D"
       THEN ER = 3:
      RETURN
8725  IF  LEFT$ (PE$,1) = "D" AND DD
       = 0 THEN ER = 2
8730  RETURN
8800  REM
8810  Z$ = B$(4):
      GOSUB 2900:
      VR$ = Z$:
      IF VR$ = "" THEN ER = 5:
      RETURN
8820  GOSUB 9500:
      IF ER <  > 0 THEN  RETURN
8825  V$ = VR$:
      Z$ = B$(5):
      GOSUB 2900:
      CF$ = Z$:
      FOR I = 1 TO 6:
         IF CF$(I) = CF$ THEN 8860
8850  NEXT I:
      ER = 7:
      RETURN
8860  Z$ = B$(6):
      GOSUB 2900:
      IF Z$ = "" THEN ER = 3:
      RETURN
8870  Z1$ =  LEFT$ (Z$,1):
      IF Z1$ =  CHR$ (34) AND VT(VR)
       <  > 1 THEN 8920
8880  IF Z1$ >  = "0" AND Z1$ <
       = "9" AND VT(VR) = 1 THEN 8920
8890  VT = VT(VR):
      VR$ = Z$:
      GOSUB 9500:
      IF ER <  > 0 THEN  RETURN
8910  IF VT <  > VT(VR) THEN ER = 8:
      RETURN
8920  IF  LEFT$ (B$(7),1) < "0" OR
       LEFT$ (B$(7),1) > "9" THEN ER
       = 4:
      RETURN
8930  ER = 0:
      PE$ = "":
      PA$(NL,1) = V$:
      PA$(NL,2) = CF$:
      PA$(NL,3) = Z$:
      Z$ = B$(7):
      GOSUB 2900:
      PA$(NL,4) = Z$:
      RETURN
9000  REM
9010  Z$ = B$(4):
      GOSUB 2900:
      VR$ = Z$:
      IF DD = 0 THEN ER = 2:
      RETURN
9020  GOSUB 9500:
      IF ER <  > 0 THEN  RETURN
9025  IF VM(VR) <  > 1 THEN ER = 6:
      RETURN
9030  Z$ = B$(5):
      GOSUB 2900:
      IF Z$ = "" THEN ER = 0:
      PE$ = "":
      PA$(NL,1) = VR$:
      RETURN
9040  CF$ = Z$:
      FOR I = 1 TO 6:
         IF CF$(I) = CF$ THEN 9060
9050  NEXT I:
      ER = 7:
      RETURN
9060  Z$ = B$(6):
      GOSUB 2900:
      IF Z$ = "" THEN ER = 8:
      RETURN
9070  PE$ = "":
      PA$(NL,1) = VR$:
      PA$(NL,2) = CF$:
      PA$(NL,3) = Z$:
      ER = 0:
      RETURN
9100  REM
9110  Z$ = B$(4):
      GOSUB 2900:
      VR$ = Z$:
      GOSUB 9500:
      IF ER <  > 0 THEN  RETURN
9125  IF VT(VR) <  > 1 THEN ER = 9:
      RETURN
9130  Z$ = B$(5):
      V1$ = VR$:
      GOSUB 2900:
      VR$ = Z$:
      IF  LEFT$ (Z$,1) >  = "0" AND
       LEFT$ (Z$,1) <  = "9" THEN 9170
9150  GOSUB 9500:
      IF ER <  > 0 THEN  RETURN
9160  IF VT(VR) <  > 1 THEN ER = 9:
      RETURN
9170  Z$ = B$(6):
      GOSUB 2900:
      OP$ = Z$:
      FOR I = 1 TO 4:
         IF OP$(I) = OP$ THEN 9185
9180  NEXT I:
      ER = 7:
      RETURN
9185  V2$ = VR$:
      Z$ = B$(7):
      GOSUB 2900:
      IF  LEFT$ (Z$,1) >  = "0" AND
       LEFT$ (Z$,1) <  = "9" THEN 9200
9190  VR$ = Z$:
      GOSUB 9500:
      IF ER <  > 0 THEN  RETURN
9195  IF VT(VR) <  > 1 THEN ER = 9:
      RETURN
9200  PE$ = "":
      PA$(NL,1) = V1$:
      PA$(NL,2) = V2$:
      PA$(NL,3) = OP$:
      PA$(NL,4) = Z$:
      ER = 0:
      RETURN
9300  REM
9310  J = 1:
      Z$ = B$(3):
      GOSUB 2900:
      IF  VAL (Z$) <  = 0 THEN ER
       = 3:
      RETURN
9320  PE$ = Z$:
      FOR ID = 4 TO 13:
         Z$ = B$(ID):
         GOSUB 2900:
         IF Z$ = "" THEN 9370
9335     IF Z$ = "FIN" THEN 9380
9340     IS =  VAL (Z$):
         IF IS <  = 0 THEN ER = 4:
         RETURN
9355     IF IS = NL OR IS$(IS) = "REPIT
         E" THEN ER = 2:
         RETURN
9360     PA$(NL,J) = Z$:
         J = J + 1
9370  NEXT ID
9380  ER = 0:
      IF J = 1 THEN ER = 3
9390  RETURN
9500  REM
9530  FOR I = 1 TO 10:
         IF VN$(I) = VR$ THEN VR
          = I:
         ER = 0:
         RETURN
9540  NEXT I:
      ER = 5:
      RETURN
10000 REM -----
10005 REM DATOS
10007 REM -----
10010 DATA 5
10020 DATA "DEFINICIONES","INSTRUCCIONE
      S","LISTADO","EJECUCION","FIN DE
      TAREA"
10030 DATA "LEE","ESC","BOR","DEC","BUS
      ","CAL","REP"
10040 DATA "=","<",">","<=",">=","<>","
      $","+","-","/"
10060 DATA "LA INSTRUCCION NO EXISTE","
      ERROR LOGICO","EL PARAMETRO NO EX
      ISTE","NUMERO LINEA NO VALIDO","V
      ARIABLE NO DEFINIDA","VARIABLE NO
       VALIDA","OPERANDO NO VALIDO","CO
      MPARACION NO VALIDA","CALCULO ILE
      GAL"
11000 REM
11010 TI$ = "PULSAR UNA TECLA":
      YT = 22:
      GOSUB 2700:
      GOSUB 1200:
      GOSUB 2800:
      RETURN
12000 REM
12030 GOSUB 500:
      POKE 34,5:
      POKE 32,1:
      POKE 33,38:
      POKE 35,23:
      VTAB 6:
      FOR IS = 1 TO 99:
         IF IS$(IS) = "" THEN 12130
12080    HTAB 3:
         PRINT IS;:
         HTAB 7:
         PRINT IS$(IS);:
         HTAB 19:
         PRINT PE$(IS);:
         FOR I = 1 TO 10:
            IF PA$(IS,I) = "" THEN 1211
            0
12100       HTAB 27:
            PRINT PA$(IS,I)
12110    NEXT I:
         IF  PEEK ( - 16384) > 128
          THEN  POKE  - 16368,0:
         GOSUB 1200
12125    PRINT
12130 NEXT IS:
      PRINT :
      PRINT :
      TI$ = "PULSAR UNA TECLA":
      YT = 22:
      GOSUB 2700:
      GOSUB 1200:
      GOSUB 2800:
      TEXT :
      RETURN
13000 REM
13030 YT = 23:
      GOSUB 2800
13040 FOR I = 1 TO 205:
         FOR T = 1 TO 100:
```

```
         NEXT T:
         CH$ =  MID$ (HE$,I,1):
         IF CH$ = "" THEN CH$ = " "
13060    A$ =  RIGHT$ (A$,37) + CH$:
         HTAB 2:
         VTAB 23:
         PRINT A$;:
         A0 =  PEEK ( - 16384):
         IF A0 > 128 THEN 13110
13100 NEXT I:
      GOTO 13040
13110 GOSUB 2800:
      HTAB 3:
      VTAB 23:
      PRINT IN$;:
      IF CO <  > 0 THEN  PRINT " ";CO;
13130 HTAB 27:
      PRINT "AYUDA.CTRL-Y":
      POKE  - 16368,0:
      RETURN
14000 REM
14025 IN$ = "EJECUCION":
      RL = 2:
      RW = MX + 1:
      FOR I = 1 TO 10:
         VA$(I) = "":
      NEXT :
      FOR NL = 1 TO 99:
         IF IS$(NL) = "" THEN 14090
14060    IS$ = IS$(NL):
         PE$ = PE$(NL):
         FOR I = 1 TO 10:
            P$(I) = PA$(NL,I):
         NEXT I:
         GOSUB 8500:
         ON IS GOSUB 14500,14800,15100,
         15400,15700,16000,16300
14090 NEXT NL:
      RETURN
14500 REM
14510 IF PE$ = "D" THEN 14650
14520 FOR I = 1 TO 10:
         IF P$(I) = "" THEN 14540
14530 NEXT I
14540 NC = I - 1:
      FOR II = 1 TO NC:
         D$(II) = P$(II):
         VR$ = D$(II):
         GOSUB 9500:
         NV(II) = VR:
         TP(II) = VT(VR):
         LC(II) = VL(VR):
         ND(II) = VD(VR):
         DX(II) = VX(VR):
         DY(II) = VY(VR):
         CX(II) = DX(II) + 11:
         CY(II) = DY(II):
      NEXT II:
      GOSUB 2000:
      GOSUB 3000:
       IF FS = 7 THEN  RETURN
14590 IF FS = 6 THEN 14540
14600 FOR I = 1 TO NC:
          VA$(NV(I)) = B$(I):
       NEXT I:
       RETURN
14650 IF RL > MX THEN  RETURN
14660 NR = RL:
       GOSUB 6000:
       GOSUB 6200:
       GOSUB 6600:
       FOR I = 1 TO 10:
          VK%(I) = 0:
       NEXT I:
       FOR ID = 1 TO 10:
          IF P$(ID) = "" THEN 14720
14690     IF VM(ID) = 0 THEN 14710
14700     VR$ = P$(ID):
          GOSUB 9500:
          VK%(VR) = 1
14710 NEXT ID
14720 FOR I = 1 TO 10:
         IF VM(I) = 0 THEN 14740
14730    IF VK%(I) = 1 THEN VA$(I)
          =  MID$ (AA$,VI(I),VL(I))
14740 NEXT I:
      RL = RL + 1:
      RETURN
14800 REM
14810 IF PE$ = "D" THEN 14950
14820 FOR I = 1 TO 10:
         IF P$(I) = "" THEN 14850
14830 NEXT I
14850 NC = I - 1:
      FOR II = 1 TO NC:
         TP(II) = 4:
         VR$ = P$(II):
         GOSUB 9500:
         D$(II) = VR$:
         LC(II) = VL(VR):
         ND(II) = VD(VR):
         DX(II) = VX(VR):
         DY(II) = VY(VR):
         CX(II) = DX(II) + 11:
         CY(II) = DY(II):
         B$(II) = VA$(VR):
      NEXT II:
      GOSUB 2000:
      HTAB 23:
      VTAB 23:
      INVERSE :
      PRINT "PULSAR UNA TECLA";:
      NORMAL :
      GOSUB 1200:
      HTAB 23:
      PRINT  LEFT$ (BA$,16);:
      RETURN
14950 GOSUB 6000:
      NR = RW - 1:
      GOSUB 6200:
      FOR I = 1 TO 10:
         VK%(I) = 0:
         IF VM(I) = 1 THEN VP$(I)
          =  MID$ (AA$,VI(I),VL(I))
14980 NEXT I:
      FOR ID = 1 TO 10:
         IF P$(ID) = "" THEN 15030
15000    IF VM(ID) = 0 THEN 15020
15010    VR$ = P$(ID):
         GOSUB 9500:
         VK%(VR) = 1
15020 NEXT ID
15030 AA$ = "":
      FOR I = 1 TO 10:
         IF VM(I) = 1 AND VK%(I)
          = 1 THEN VA$(I) =  LEFT$ (VA$
         (I),VL(I)):
         AA$ = AA$ + VA$(I) +  MID$ (BA
         $,1,VL(I) -  LEN (VA$(I)))
15050    IF VM(I) = 1 AND VK%(I)
          = 0 THEN AA$ = AA$ + VP$(I)
15060 NEXT I:
      NR = RW:
      GOSUB 6400:
      NR = 1:
      MX = MX + 1:
      AA$ =  STR$ (MX):
      GOSUB 6400:
      GOSUB 6600:
      RW = RW + 1:
      RETURN
15100 REM
15110 IF PE$ <  > "D" THEN  GOSUB 500:
      RETURN
15120 RL = 2:
      RW = 2:
      MX = 1:
      GOSUB 6000:
      NR = 1:
      AA$ = "1":
      GOSUB 6400:
      GOSUB 6600:
      RETURN
15400 REM
15410 VR$ = P$(1):
      GOSUB 9500:
      V1$ = VA$(VR):
      Z$ = V1$:
      GOSUB 2900:
      V1$ = Z$:
      N = VT(VR):
      IF  LEFT$ (P$(3),1) <  >  CHR$ (3
      4) THEN 15480
15440 L =  LEN (P$(3)):
      FOR I = 2 TO L:
         IF  MID$ (P$(3),I,1) =
          CHR$ (34) THEN 15460
15450 NEXT I
15460 V2$ =  MID$ (P$(3),2,I - 2):
      Z$ = V2$:
      GOSUB 2900:
      V2$ = Z$:
      GOTO 15520
15480 IF  LEFT$ (P$(3),1) >  = "0"
       AND  LEFT$ (P$(3),1) <  = "9"
       THEN V2$ = P$(3):
      GOTO 15510
15490 VR$ = P$(3):
      GOSUB 9500:
      V2$ = VA$(VR):
      Z$ = V2$:
      GOSUB 2900:
      V2$ = Z$
15510 IF N = 1 THEN V1 =  VAL (V1$):
      V2 =  VAL (V2$)
15520 FOR I = 1 TO 6:
         CF(I) = 0:
         IF CF$(I) = P$(2) THEN CF(I)
          = 1
15530 NEXT I:
      IF N <  > 1 THEN 15540
15535 IF (V1 = V2 OR CF(1) = 0) AND (V1
       < V2 OR CF(2) = 0) AND (V1
       > V2 OR CF(3) = 0) AND (V1
       <  = V2 OR CF(4) = 0) AND (V1
       >  = V2 OR CF(5) = 0) AND (V1
       <  > V2 OR CF(6) = 0) THEN 15560

15537 RETURN
15540 IF (V1$ = V2$ OR CF(1) = 0)
       AND (V1$ < V2$ OR CF(2) = 0)
       AND (V1$ > V2$ OR CF(3) = 0)
       AND (V1$ <  = V2$ OR CF(4)
       = 0) AND (V1$ >  = V2$ OR CF(5)
       = 0) AND (V1$ <  > V2$ OR CF(6)
       = 0) THEN 15560
15550 RETURN
15560 NI =  VAL (P$(4)):
      IF IS$(NI) = "" THEN  RETURN
15580 IS$ = IS$(NI):
      PE$ = PE$(NI):
      FOR I = 1 TO 10:
         P$(I) = PA$(NI,I):
      NEXT I:
      GOSUB 8500:
      ON IS GOSUB 14500,14800,15100,500
      00,15700,16000,16300:
      RETURN
15700 REM
15710 VR$ = P$(1):
      GOSUB 9500:
      Z$ = VA$(VR):
      GOSUB 2900:

      IF P$(2) = "=" THEN CH$ = P$(3)
15740 GOSUB 6000:
      FOR IR = 2 TO MX:
```

```
         NR = IR:
         GOSUB 6200:
         Z$ =  MID$ (AA$,VI(VR),VL(VR))
         :
         GOSUB 2900:
         IF Z$ <  > CH$ THEN 15830
15800    RL = IR:
         FOR I = 1 TO 10:
            IF VM(I) = 1 THEN VA$(I)
               =  MID$ (AA$,VI(I),VL(I))
15820    NEXT I:
         RETURN
15830 NEXT IR:
      FOR I = 1 TO 10:
         IF VM(I) = 1 THEN VA$(I)
            = ""
15850 NEXT I:
      RETURN
16000 REM
16010 IF  LEFT$ (P$(2),1) >  = "0"
       AND  LEFT$ (P$(2),1) <  = "9"
       THEN V1 =  VAL (P$(2)):
      GOTO 16040
16020 VR$ = P$(2):
      GOSUB 9500:
      V1 =  VAL (VA$(VR))
16040 IF  LEFT$ (P$(4),1) >  = "0"
       AND  LEFT$ (P$(4),1) <  = "9"
       THEN V2 =  VAL (P$(4)):
      GOTO 16070
16050 VR$ = P$(4):
      GOSUB 9500:
      V2 =  VAL (VA$(VR))
16070 OP$ = P$(3):
      FOR I = 1 TO 4:
         CF(I) = 0:
         IF OP$ = OP$(I) THEN CF(I)
          = 1:
         GOTO 16090
16080 NEXT I
16090 VV = (V1 * V2) * CF(1) + (V1
       + V2) * CF(2) + (V1 - V2)
       * CF(3) + (V1 / V2) * CF(4):
      VR$ = P$(1):
      GOSUB 9500:
      VA$(VR) =  STR$ (VV):
      RETURN
16300 REM
16310 VT =  VAL (PE$):
      FOR I = 1 TO 10:
         IF P$(I) = "" OR P$(I) = "FIN"
          THEN 16340
16330    PP$(I) = P$(I):
      NEXT I
16340 FT = I - 1:
      IZ = 1:
      IT = 1
16360 NI =  VAL (PP$(IZ)):
      IF IS$(NI) = "" THEN 16400
16380 IS$ = IS$(NI):
      PE$ = PE$(NI):
      FOR I = 1 TO 10:
         P$(I) = PA$(NI,I):
      NEXT I:
      GOSUB 8500:
      ON IS GOSUB 14500,14800,15100,154
      00,15700,16000,50000
16400 IZ = IZ + 1:
      IF IZ <  = FT THEN 16360
16410 IT = IT + 1:
      IF IT <  = VT THEN IZ = 1:
      GOTO 16360
16420 RETURN
30000 REM
30030 YT = 21:
      TI$ = ER$(ER):
      INVERSE :
      GOSUB 2700:
      PRINT BP$;:
      GOSUB 1200:
      NORMAL :
      GOSUB 2800:
      IF NL > 0 THEN IS$(NL) = ""
30070 RETURN
40000 REM
40030 IF  PEEK (222) = 5 THEN 5540
40040 GOSUB 500:
      YT = 13:
      TI$ = "ERROR EN FASE DE EJECUCION
      ":
      GOSUB 2700:
      GOSUB 11000:
      GOTO 140
50000 RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - APPLE

MP$	= nombre programa	**DY(*)**	= coordenadas Y descripciones
OP	= opción elegida en menú principal	**CX(*)**	= coord. X campos
NV	= número de voces	**CY(*)**	= coord. Y campos
OP$(*)	= operadores matemáticos (+,−,*,/)	**IS$(*)**	= instrucciones del programa
BA$	= 40 blanks	**PA$(*)**	= parámetros asociados a la instrucción
BU$	= 40 guiones		
BE$	= 40 iguales	**PE$(*)**	= periféricos asociados a la instrucción
HE$	= cadena de la ayuda		
B$(*)	= buffer data entry	**ER$(*)**	= cadena con los mensajes de error
TP(*)	= tipos de los campos		
ND(*)	= número decimales	**ER**	= código error
LC(*)	= longitudes campos	**AA$**	= buffer del disco
D$(*)	= descripciones campos	**NF$**	= nombre fichero
		LR	= longitud registro
DX(*)	= coordenadas X descripciones	**NR**	= número registro

El ordenador para el ordenador/Versión Philips VG 8010

```
10   REM ------------------------------
20   REM EL ORDENADOR PARA EL ORDENADOR
25   REM      Version MSX  (cinta)
30   REM ------------------------------
31   CLEAR 7500:DEFINTA-V:ONERRGOTO15000:ONSTOP GOSUB21000:STOP ON
32   KEY OFF:KEY 1,CHR$(15):KEY 2,CHR$(16):KEY 3,CHR$(17):KEY 4,CHR$(19):KEY 5,CHR$(20):WIDTH 40
33  CLS:LOCATE 3,7:PRINT"Pulsar la tecla 'CAPS' y asegurarse":LOCATE 2,10:PRINT"de que la luz am
arilla este encendida.":LOCATE8,16:PRINT"Despues pulsar una tecla."
35   IF INKEY$=""THEN 35 ELSE GOSUB 600
40   DIMB$(60),D$(60),B2$(60),OP$(11)
50   DIMTF(15),TP(60),LC(60),ND(60),CX(60),CY(60),DX(60),DY(60)
60   M3=15:DIM IST$(M3),SO(M3),NT(M3),PER$(M3),PAR$(M3,10),TD$(15)
70   A1=1:A2=31:NS$="=":AS$="-":PD$=".":BL$=" ":ER$="E":MP$="| MENU PRINCIPAL |"
71   BP$="O6L60ER16ER16ER16CCC"
75   F$=CHR$(1)+CHR$(87)
77   Q$="":FOR I=0 TO 39:Q$=Q$+F$:NEXT I
80   SZ=2:MI=1
90   X0=16:Y0=1:XC=5:YC=8:XS=3
100  US$=CHR$(13):ZE$=CHR$(48):GIU$=CHR$(32)
110  HE$="CTRL Y (AYUDA)"
115  H1$="E.G.S.            Ediciones Forum, S.A."
120  REM
130  RESTORE
135  FOR I=1 TO 11:READ OP$(I):NEXT
140  READ PROG$
150  READ NV
160  FOR I=1 TO NV:READ VO$(I):NEXT
170  MA=NV+1
180  VO$(MA)="Fin de tarea"
190  FOR I=1 TO 15
200  READ TF(I)
210  NEXT I
220  REM MAIN
230  GOSUB1300
240  IF OPZ=MA THEN FOR I=1 TO 1000:NEXT:CLS:COLOR15,4:END
250  LM=OPZ-1:GOSUB7000
260  GOSUB 2000
270  GOSUB 3000
280  IF FS=6 THEN 260
290  GOTO 230
300  REM
310  REM - DEFINICION FICHERO -
323  GOSUB 600:G9=3
325  FOR I=1 TO 60:B2$(I)=B$(I):B$(I)="":NEXT
326  FOR I=1 TO 60 STEP6:IF (LEFT$(B2$(I),1)<>" "AND LEFT$(B2$(I),1)<>"")AND B2$(I+1)<>"1"THEN B
2$(I+1)="3"
327  IF VAL(B2$(I+1))=1 AND VAL(B2$(I+2))<=VAL(B2$(I+3)) THEN B2$(I+2)=STR$(VAL(B2$(I+3))+1)
328  NEXT I
330  OPZ=2:GOSUB 7000:GOSUB 2000
338  JJ=1
340  FOR I=1 TO 5:Y=13+I
342  FOR J=0 TO 1:X=2+18*J
344  GOSUB 11000:IF RIGHT$(B2$(JJ),1)="-" THEN B2$(JJ)=""
345  PRINT B2$(JJ):JJ=JJ+6
346  NEXT J,I:JJ=0
350  LOCATE0,12
360  PRINT"Introducir 'S' para campos a memorizar"
370  LOCATE 0,13:PRINT Q$;
400  GOSUB 3000:IF FS=7 THEN 230 ELSE IF FS=6 THEN 330
402  IF VAL(B$(2))>15 THEN 330
403  CLS:LOCATE 5,8:PRINT"Si la unidad de cinta va a":LOCATE 4,11:PRINT"utilizarse, pulsar la te
cla 'S'."
405  LOCATE 9,16:PRINT"Si no, la tecla 'N'."
407  O$=INKEY$:IF O$="S" OR O$="N" THEN409 ELSE 407
409  IF O$="N"THEN 499
410  CLS:LOCATE 4,8:PRINT"Si el fichero ya existe,pulsar 'S',":LOCATE 6,13:PRINT"se debe crear p
ulsar 'N'."
411  O$=INKEY$:IF O$="S" OR O$="N" THEN 413 ELSE 411
413  NM$=B$(1):NR=VAL(B$(2))
420  LR=0:FOR I=3 TO 12:IF LEFT$(B$(I),1)="S" THEN LR=LR+VAL(B2$((I-3)*6+3)):SK$(I-2)="S" ELSE S
K$(I-2)=""
430  NEXT I:IF O$="N" THEN 499
436  GOSUB 640:GOSUB 600
440  OPEN "CAS:"+NM$ FOR INPUT AS #1
450  FOR I=1 TO NR:INPUT #1,TD$(I):NEXT:CLOSE #1
499  REM
500  GOTO 230
600  REM --- ESPERAR -----
610  CLS:LOCATE 16,11:PRINT"ESPERAR"
620  RETURN
640  REM --- POSIC. CINTA LEE ---
650  CLS:LOCATE 1,10:PRINT"Posicionar la cinta y apretar PLAY."
660 LOCATE 6,14:PRINT"Despues pulsar una tecla."
670  IF INKEY$="" THEN 670
675  RETURN
700  REM ------------------
705  REM FASE INSTRUCCIONES
720  RESTORE 10160
740  FOR I=1 TO 8:READ FR$(I),NP(I):NEXT
744  G5=1
745  FOR NI=G5 TO M3
750  OPZ=3:GOSUB 7000:FOR I=1 TO 3:DY(I)=CY(I)-2:NEXT
760  GOSUB 2000:GOSUB 3000:IF FS=7 THEN IF NI>1 GOTO 950 ELSE 230
762  IF FS=6 THEN 760
765  IF LEFT$(B$(2),3)="   " THEN SI=9:GOTO 780
770  FOR SI=1 TO 8:IF LEFT$(B$(2),3)=FR$(SI) THEN SO(NI)=SI:GOTO 780
772  NEXT SI
775  CLS:PRINT"ERROR":FOR I=1 TO 1000:NEXT:GOTO 760
780  ON SI GOSUB 800,800,830,880,840,880,850,870,860
790  GOTO 900
799  REM --- LEE --- ESCRIBE ---
800  J$=LEFT$(B$(3),1):IF J$="V" OR J$="N" THEN 801 ELSE GOTO 813
801  IF LEFT$(B$(4),3)="   " THEN 813
802  FOR J=4 TO 3+NP(SI)
803  IF LEFT$(B$(J),3)="   " THEN 815
804  FOR I=1 TO 60 STEP 6
805  IF B2$(I)=B$(J) THEN 810
807  NEXT I:GOTO 813
810  NEXT J:GOTO 815
813  CLS:LOCATE 3,10:PRINT"ERROR, REPETIR LA INSTRUCCION":FOR I=1 TO 1500:NEXT:RETURN 760
815  CLS:LOCATE 10,10:PRINT "C O R R E C T O"
819  FOR I=1 TO 1000:NEXT:RETURN
830  REM --- BORRA ---
832  B$(3)=LEFT$(B$(3),1):IF B$(3)="V" OR B$(3)="N" THEN 835
833  GOTO 813
835  GOSUB 815:RETURN
840  J=4:REM --- CALCULA ---
841  FOR I=1 TO 60 STEP 6:IF B2$(I)=B$(J) THEN 842 ELSE NEXT I:GOTO 813
842  IF B2$(I+1)="3" THEN 813
843  IF J=7 THEN 846
844  IF J=5 THEN J=7 ELSE J=5
845  IFASC(B$(J))<48 OR ASC(B$(J))>57 THEN 841 ELSE 843
846  FOR I=1 TO 5:IF LEFT$(B$(6),1)=OP$(I) THEN 848
847  NEXT I:GOTO 813
848  GOSUB 815:RETURN
850  REM --- REPITE ----
851  FOR I=4 TO 13
854  IF LEFT$(B$(I),3)="FIN" THEN 859
855  IF VAL(B$(I))=VAL(B$(1)) THEN 813
856  IF VAL(B$(I))>99 THEN 813
858  NEXT I:GOTO 813
859  GOSUB 815:RETURN
860  REM --- BORRA INSTRUCCION ---
861  FOR I=1 TO NI:IF NT(I)=VAL(B$(1)) THEN NT(I)=0:IST$(I)="":B$(1)="0"
863  NEXT I
864  RETURN
870  RETURN
880  REM --- BUSCA --- DECIDE ---
882  FOR I=1 TO 60 STEP 6:IF B2$(I)=B$(4) THEN 884
883  NEXT I:GOTO 813
884  IF LEFT$(B$(5),3)="   " AND SI=4 THEN 898
888  FOR I=6 TO 11:IF LEFT$(B$(5),2)=OP$(I) THEN 894 ELSE NEXT I:GOTO 813
```

```
894 IF SI=4 THEN 898
895 FOR I=1 TO 99:IF VAL(B$(7))=I THEN 898
896 NEXT I:GOTO 813
898 GOSUB 815:RETURN
900 IST$(NI)=B$(2):NT(NI)=VAL(B$(1))
905 IF LEFT$(IST$(NI),3)=FR$(8) THEN NT(NI)=99:GOSUB 600:GOTO 950
910 PER$(NI)=LEFT$(B$(3),1)
915 IF LEFT$(IST$(NI),3)="REP" THEN PER$(NI)=B$(3)+SPACE$(4)
920 FOR I=1 TO NP(SI):PAR$(NI,I)=B$(I+3)
922 IF LEFT$(PAR$(NI,I),3)="   " THEN PAR$(NI,I)=""
925 NEXT I
930 NEXT NI:NI=NI-1
950 FOR I=0 TO NI-1
955 IF NT(I)>NT(I+1)THENN1=NT(I):NT(I)=NT(I+1):NT(I+1)=N1:I1$=IST$(I):IST$(I)=IST$(I+1):IST$(I+
1)=I1$:N1=SO(I):SO(I)=SO(I+1):SO(I+1)=N1:GOTO 960
956 IF NT(I)=NT(I+1) THEN 957 ELSE 959
957 FOR J=I TO NI-1:NT(J)=NT(J+1):IST$(J)=IST$(J+1):PER$(J)=PER$(J+1):SO(J)=SO(J+1)
958 FOR II=1 TO 10:PAR$(J,II)=PAR$(J+1,II):NEXTII:NEXTJ:IST$(NI)="     ":NI=NI-1:GOTO950
959 NEXT I:G5=NI+1:GOTO 230
960 I1$=PER$(I):PER$(I)=PER$(I+1):PER$(I+1)=I1$:FORII=1TO10:I1$(II)=PAR$(I,II):PAR$(I,II)=PAR$(
I+1,II):PAR$(I+1,II)=I1$(II):NEXTII
962 GOTO 950
990 CLS:FOR I=1 TO G5-1:PRINTNT(I);TAB(4)IST$(I);"  ";PER$(I);:FORII=1TONP(SO(I)):IF PAR$(I,II)
="" THEN 994 ELSE PRINT TAB(27)PAR$(I,II):NEXT II
994 PRINT:PRINT:NEXT I
996 PRINT"Pulsar una tecla para volver al menu"
997 IF INKEY$="" THEN 997
998 GOTO 230
1300 REM ----
1310 REM MENU
1322 CLS:COLOR 15,4
1335 LOCATE1,0:PRINT H1$
1340 LOCATE2,2:PRINT"┌──────────────────────────┐":LOCATE2,3:PRINT"│";PROG$;"│";:LOCATE2,4
:PRINT"└──────────────────────────┘"
1345 LOCATE11,6:PRINTMP$;:LOCATE11,5:PRINT"┌──────────┐";
1346 LOCATE11,7:PRINT"└──────────┘";
1350 FOR I=MI TO MA
1360 Y=I*SZ+YC:X=XC:GOSUB 11000
1370 II=I
1380 IF II=MA THEN II=0
1390 PRINT II;"-";VO$(I)
1400 NEXT
1403 LOCATE1,23:PRINT "Posicionar el cursor y pulsar RETURN";
1410 I=MI
1420 Y=I*SZ+YC:X=XS:GOSUB 11000
1430 SP=YC-6
1440 PRINT"->";
1450 OPZ$=INKEY$:IF OPZ$="" THEN 1450
1455 OPZ=VAL(OPZ$):IF OPZ$=US$ THEN 1530
1460 IF OPZ>4 THEN 1450
1470 IF OPZ$=ZE$ THEN X=XS:GOSUB 11000:PRINTSPC( SP);:I=MA:GOTO 1420
1480 IF ASC(OPZ$)>48 AND ASC(OPZ$)<58 THENX=XS:GOSUB 11000:PRINTSPC( SP);:I=OPZ:GOTO 1420
1490 IF OPZ$<>GIU$ THEN 1450
1500 X=XS:GOSUB 11000:PRINTSPC( SP);
1510 I=I+1:IF I>(MA) THEN 1410
1520 GOTO 1420
1530 REM
1540 II=I:IF I>4 THEN II=0
1544 X=X+2
1545 FOR J=1 TO 6:GOSUB 11000:FOR IJ=1 TO 30:NEXT IJ
1550 PRINTSPACE$(30):GOSUB 11000:FOR IJ=1 TO 30:NEXT IJ
1560 PRINTII;"-";VO$(I):NEXTJ
1580 OPZ=I
1581 IF OPZ=4 THEN RETURN 4400
1583 IF OPZ=2 AND G5>1 THEN RETURN 745
1585 IF OPZ=2 THEN RETURN 700
1587 IF OPZ=3 THEN RETURN 990
1588 IF OPZ=1 AND(G5>1 OR G9>1) THEN 1600
1590 RETURN
1600 GOSUB 7000:FOR I=1 TO 60:TP(I)=4:B$(I)=B2$(I):NEXT:GOSUB 2000:GOSUB 7000:GOSUB 3000:IF FS=7
 THEN RETURN 230 ELSE RETURN 1600
2000 REM
2010 REM --- MASCARA ---
2030 CLS:COLOR 12,14
2050 IF OPZ>4 THEN LOCATE1,0:PRINTH1$:GOTO 2240
2120 IF OPZ>2 THEN GJ=2 ELSE GJ=1
2130 L=LEN(VO$(GJ))
2140 FM=(38-L)/2:FM$=SPACE$(FM)
2150 LOCATE1,0:PRINTH1$;
2180 IF OPZ=1 GOTO 12000
2185 IF OPZ=3 THEN 2215
2190 LOCATE14,5:PRINT"┌───────┐";
2195 LOCATE14,6:PRINT"  FASE";OPZ;:LOCATE 22,6:PRINT "│"
2200 LOCATE 14,7:PRINT"└───────┘";
2215 LOCATE 0,3:PRINT FM$;" ";VO$(GJ);" ";FM$:LOCATE FM,2
2216 G8$=LEFT$(O$,L*2):PRINT CHR$(1);CHR$(&H58);G8$;CHR$(1);CHR$(&H59):LOCATE FM,4:PRINT CHR$(1)
;CHR$(&H5A);G8$;CHR$(1);CHR$(&H5B);
2218 LOCATE 0,21:PRINT Q$:LOCATE 25,22:PRINT HE$
2240 FOR J=1 TO NUC
2242 IF TP(J)=0 THEN 2420
2245 A$=""
2270 LOCATEDX(J),DY(J):PRINTD$(J)
2275 IF TP(J)<>5 AND TP(J)<>4 THEN 2290
2280 LOCATECX(J),CY(J)
2285 PRINT B$(J):GOTO 2420
2290 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J)=2 THEN C$=NS$:GOTO 2310
2300 C$=AS$
2310 A$=STRING$(N1-N2-1,C$)
2320 IF N2=0 THEN A$=A$+C$:GOTO 2350
2330 A$=A$+PD$
2340 FOR I=1 TO N2:A$=A$+C$:NEXT I
2350 LOCATECX(J),CY(J)
2360 PRINT A$
2380 B$(J)=SPACE$(N1-N2-1)
2410 B$(J)=B$(J)+RIGHT$(A$,N2+1)
2420 NEXT J
2430 RETURN
3000 REM -------
3010 REM D.ENTRY
3020 B7=0
3030 J=1
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 PLAYBP$:FS=0:J=NUC:GOTO3040
3150 IF J<NUC THEN J=J+1:GOTO 3040
3160 PLAYBP$:FS=0:GOTO3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 PLAYBP$:FS=0:GOTO3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB 11000:GOTO 3060
3200 PLAYBP$:FS=0:GOTO3040
3210 IFFS>8THENFS=0:GOTO3060
3220 IFFS=8THENFS=0:GOTO3100
3230 RETURN
3240 REM ------------------
3250 REM LECTURA DEL CAMPO
3270 FS=0
3275 IF TP(NC)=0 THEN B7=B7+1:GOTO 3276 ELSE 3280
3276 IF B7<38 THEN RETURN ELSE ERR
3280 IF TP(NC)=4 THEN RETURN
3290 IF TP(NC)=2 THEN TP(NC)=1
3300 IF MID$(B$(NC),CC,1)=PD$ THEN X=X+S:CC=CC+S
3310 GOSUB 11000
3320 A$=INKEY$:IFA$=""THENPRINT"█";:GOSUB 11000:PRINTMID$(B$(NC),CC,1);:GOSUB 11000:GOTO 3320
3330 CA=ASC(A$):IFCA>A2THEN3380
3340 FOR I=1 TO 15
3350 IF TF(I)=CA THEN FS=I:GOSUB 3630:RETURN
3360 NEXT I
3370 GOTO 3320
3380 IF TP(NC)<>1 THEN 3440
3390 IF CA=45 OR CA=46 OR (CA>=48 AND CA<=57) THEN 3410
3400 PLAYBP$:GOTO 3320
3410 IF CA<>46 THEN 3440
3420 GOSUB 4270
3430 IF SWP=1 THEN 3400
3440 S=1: GOSUB 3520
3450 IF CC=LC(NC) THEN RETURN
3460 CC=CC+1:X=X+1:GOTO 3300
3520 REM ----------------
3530 REM TRANSF.AL BUFFER
3550 IF CC=1 THEN SN$= "":GOTO 3570
```

```
3560 SN$=LEFT$(B$(NC),CC-1)
3570 IF CC=LC(NC) THEN DS$= "":GOTO 3590
3580 N=LC(NC)-CC:DS$=RIGHT$ (B$(NC),N)
3590 B$(NC)=SN$+A$+DS$
3620 PRINT A$
3622 RETURN
3630 REM --------
3640 REM COMANDOS
3650 REM --------
3655 IF FS=5 AND OPZ=1 THEN 300
3657 IF FS=5 THEN RETURN
3660 IF FS>5 THEN 3690
3680 RETURN
3690 IF FS=9 THEN GOSUB 13000:RETURN
3700 IF FS=8 THEN GOSUB 4080:RETURN
3710 IF FS<8 THEN RETURN
3720 ON FS-9 GOSUB 3740,3910
3730 RETURN
3740 REM -------
3750 REM INTROD.
3760 REM -------
3770 IFCC=LC(NC)THENPLAYBP$:RETURN
3780 IFCC=1THENSN$="":GOTO3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DS$=RIGHT$(B$(NC),N)
3810 IF TP(NC)=1ORTP(NC)=2THENC$=BL$:GOTO3830
3820 C$=BL$
3830 A$=SN$+C$+DS$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB11000
3870 X=XX:Y=YY
3880 PRINTB$(NC):S=-1
3890 IFRIGHT$(B$(NC),1)=PD$THEN3780
3900 RETURN
3910 REM -------
3920 REM BORRADO
3930 REM -------
3940 IF TP(NC)<>1 THEN 3960
3950 IF MID$(B$(NC),2,1)=PD$ AND CC=1 THEN RETURN
3960 IF CC<1 THEN PRINTBP$:RETURN
3970 IF CC=LC(NC) THEN DS$="":GOTO 4000
3980 N=LC(NC)-CC:DS$=RIGHT$(B$(NC),N)
3990 IF CC=1 THEN SN$="":GOTO 4010
4000 SN$=LEFT$(B$(NC),CC-1)
4010 IFTP(NC)=1ORTP(NC)=2THENC$=NS$:GOTO4030
4020 C$=AS$
4030 B$(NC)=SN$+DS$+C$
4040 XX=X:YY=Y
4050 X=CX(NC):Y=CY(NC):GOSUB 11000
4060 X=XX:Y=YY
4070 PRINTB$(NC);:S=1:RETURN
4080 REM -----
4090 REM ALIN.
4100 REM -----
4110 IFLC(NC)=1THENRETURN
4120 N1=LC(NC)-CC+1
4130 IFCC=1THENB$(NC)="":GOTO4150
4140 B$(NC)=LEFT$(B$(NC),CC-1)
4150 IFTP(NC)=1ORTP(NC)=2THEN4200
4160 FOR I=1 TO N1
4170 B$(NC)=B$(NC)+BL$
4180 NEXT I
4190 GOTO 4240
4200 FOR I=1 TO N1
4210 B$(NC)=BL$+B$(NC)
4220 NEXTI
4230 IFRIGHT$(B$(NC),1)=PD$THENB$(NC)=BL$+LEFT$(B$(NC),LC(NC)-1)
4240 X=CX(NC):Y=CY(NC)
4250 GOSUB 11000
4260 PRINT B$(NC):RETURN
4270 REM ----------
4280 REM CTRL.PUNTO
4300 SWP=0
4310 IFCC=1ORCC=LC(NC)THENSWP=1:RETURN
4320 FORI=1TOLC(NC)
4330 IFMID$(B$(NC),I,1)=PD$THENI=LC(NC):SWP=1:RETURN
4340 NEXTI:RETURN
4400 REM ----------
4401 REM DESARROLLO
4402 REM ----------
4403 RE=0:R8=0:H6=0
4405 FOR I=1 TO 10:B$(I)="":NEXTI
4410 FOR NI=1 TO M3
4420 IST$=IST$(NI):PER$=PER$(NI):NT=NT(NI)
4430 FOR I=1 TO 10:P7$(I)=PAR$(NI,I):NEXTI
4440 FOR I=1 TO 8:IFLEFT$(IST$,3)=FR$(I)THEN IO=I:GOTO 4460
4450 NEXTI:NI=NI-1:GOTO4530
4460 ONIGOSUB5000,5200,5400,5600,5800,6000,6200,6400
4500 NEXT NI:NI=NI-1
4530 LOCATE3,22:PRINT"Pulsar una tecla y vuelve al menu"
4540 IF INKEY$="" THEN 4540 ELSE 230
4600 REM ----------
4601 REM GEST.CINTA
4602 REM ----------
4645 IF IO=2 THEN 4700
4650 RE=1+RE
4655 A$=TD$(RE)
4660 K1=1:FORI=1TO10:K2=VAL(B2$((I-1)*6+3))
4670 IFK2=0ORSK$(I)<>"S"THEN4690
4680 B$(I)=MID$(A$,K1,K2)
4690 K1=K1+K2:NEXTI
4695 RETURN
4700 R8=R8+1:A$="":FOR I=1 TO 10
4710 IF VAL(B2$((I-1)*6+3))=0 OR SK$(I)<>"S" THEN 4730
4720 A$=A$+B$(I)
4730 NEXT I
4740 TD$(R8)=A$
4770 RETURN
5000 REM --- LEE ---
5008 OPZ=6:NUC=10
5010 GOSUB6800
5015 IF PER$="C" THEN GOSUB 4600:RETURN
5020 IF PER$="V" THEN GOSUB 2000:GOSUB 3000:RETURN
5030 RETURN
5200 REM --- ESCRIBE ---
5210 NUC=10:OPZ=6
5240 GOSUB 6800
5250 IF PER$="C" THEN GOSUB 4600:RETURN
5260 FOR I=1 TO 10:IF TP(I)<>0 THEN TP(I)=4
5265 NEXT I
5270 GOSUB 2000
5273 X=0:Y=23:GOSUB 11000:PRINT"Pulsar una tecla para proseguir la ejec.";
5275 IF INKEY$="" THEN 5275
5280 RETURN
5399 REM -----
5400 REM BORRA
5401 REM -----
5410 IF PER$="V" THEN CLS:RETURN
5420 IF PER$<>"C" THEN 5520
5440 FOR I=1 TO NR:TD$(I)=SPACE$(LR):NEXT:H6=3
5520 RETURN
5600 REM --- BUSCA ---
5610 K=1:GOSUB6900:P2=VAL(B2$((J-1)*6+3))
5615 IFP7$(2)=""THENOP=10ELSEGOTO5630
5620 IFASC(B$(J))>47ANDASC(B$(J))<58ORASC(B$(J))=32THENZ=VAL(B$(J)):CP$=""ELSECP$=B$(J)
5625 GOTO 5660
5630 FOR OP=6 TO 11:IF LEFT$(P7$(2),2)=OP$(OP) THEN 5640 ELSE NEXT OP:ER=5:STOP
5640 IF ASC(P7$(3))>47 AND ASC(P7$(3))<58THENZ=VAL(P7$(3)):CP$=""ELSECP$=P7$(3)
5660 IF J=1 THEN P1=1:GOTO 5700
5665 P1=1:FORI=1TOJ-1:IFSK$(I)<>"S"THEN5680
5670 P1=P1+VAL(B2$((I-1)*6+3))
5680 NEXT I
5700 FOR II=1 TO NR
5710 A$=TD$(II)
5720 GOSUB 6510:IF FC=1 THEN 5730
5725 NEXT II:CLS:LOCATE10,10:PRINT P7$(1);"NO HALLADO":GOTO 5750
5730 GOSUB 4660
5740 CLS:LOCATE10,10:PRINT"HALLADO ";P7$(1)
5750 FOR I=1 TO 1000:NEXT:RETURN
5800 REM --- CALCULA ---
5820 IF ASC(P7$(2))<48ORASC(P7$(2))>57THENK=2:GOSUB 6900:Z1=VAL(B$(J)) ELSE Z1=VAL(P7$(2))
5830 IF ASC(P7$(4))<48ORASC(P7$(4))>57THENK=4:GOSUB6900:Z2=VAL(B$(J))ELSEZ2=VAL(P7$(4))
5880 FOR I=1 TO5:IF LEFT$(P7$(3),1)=OP$(I)THEN5890 ELSE NEXTI:ER=5:GOTO 20000
5890 ON I GOSUB 5900,5905,5910,5915,5920
5893 K=1:GOSUB 6900:B$(J)=STR$(Z3):RETURN
5900 Z3=Z1+Z2:RETURN
5905 Z3=Z1-Z2:RETURN
```

```
5910 Z3=Z1*Z2:RETURN
5915 Z3=Z1/Z2:RETURN
5920 Z3=Z1^Z2:RETURN
6000 REM --- DECIDE ---
6005 K=1:GOSUB 6900:A$=B$(J)
6020 IF ASC(P7$(3))>47 AND ASC(P7$(3))<58 THEN Z=VAL(P7$(3)):CP$="":GOTO 6060
6030 IF LEFT$(P7$(3),1)=CHR$(34)THENP1=2:GOTO6035 ELSE 6040
6035 FOR I=2 TO 10:IF MID$(P7$(3),I,1)=CHR$(34) THEN P2=I-2:GOTO 6045
6037 NEXT I:P2=9:GOTO 6045
6040 K=3:GOSUB 6900:CP$=B$(J)
6042 GOTO 6060
6045 CP$=MID$(P7$(3),P1,P2)
6060 FOR OP=6 TO 11:IF LEFT$(P7$(2),2)=OP$(OP)THEN6080
6070 NEXT OP:ER=4:GOTO 20000
6080 GOSUB 6500
6090 IF FC=0 THEN 6180
6100 FOR I=1 TO M3
6110 IFVAL(P7$(4))=NT(I)GOTO6130
6115 NEXTI:P5$=P7$(4)
6120 CLS:LOCATE4,9:PRINT"LA INSTRUCCION ";P5$;" NON EXISTE":LOCATE1,12:PRINT"Pulsar una tecla pa
ra volver al menu"
6125 IF INKEY$="" THEN 6125 ELSE RETURN 230
6130 IST$=IST$(I):PER$=PER$(I):NT=NT(I)
6140 FOR K=1 TO 10:P7$(K)=PAR$(I,K):NEXT K
6150 FOR I=1 TO 8:IF LEFT$(IST$,3)=FR$(I) THEN 6170
6160 NEXT I:ER=1:GOTO 20000
6170 ON I GOSUB 5000,5200,5400,5600,5800,6000,6200,6400
6180 RETURN
6200 REM ------
6201 REM REPITE
6210 IFASC(PER$)>47ANDASC(PER$)<58THENN=VAL(PER$)ELSEK=0:P7$(K)=PER$:GOSUB6900:N=VAL(B$(J))
6220 FOR WI=1 TO 10:PP$(WI)=P7$(WI):NEXT
6230 FOR WI=1 TO N:REM inicio bucle
6240 FOR WJ=1 TO 10:IF LEFT$(PP$(WJ),3)="FIN" THEN 6300
6250 FOR WK=1 TO M3:IF VAL(PP$(WJ))=NT(WK)THEN 6252 ELSE NEXT WK:P5$=PP$(WJ):GOTO 6120
6252 IST$=IST$(WK):PER$=PER$(WK):NT=NT(WK)
6253 IF LEFT$(IST$,3)="REP "THEN CLS:LOCATE1,12:PRINT"LA INSTRUCCION NO SE PUEDE REPETIR ";NT:EL
SE GOTO 6257
6254 IF INKEY$=""THEN LOCATE9,14:PRINT"<Pulsar una tecla>":GOTO 6254
6255 RETURN 230
6257 FOR WW=1 TO 10:P7$(WW)=PAR$(WK,WW):NEXT WW
6260 FOR IK=1 TO 8:IF LEFT$(IST$,3)=FR$(IK)THEN6265 ELSE NEXTIK:ER=1:GOTO 20000
6265 IO=IK
6270 ON IK GOSUB 5000,5200,5400,5600,5800,6000,6200,6400
6280 NEXT WJ
6300 NEXT WI
6310 RETURN
6400 REM --- FIN ---
6410 IF R8=0 AND H6<>3 THEN 6440
6412 CLS:LOCATE1,10:PRINT"Posicionar la cinta y apretar REC y PLAY":LOCATE9,15:PRINT"despues pul
sa una tecla"
6415 IF INKEY$=""GOTO6415
6420 GOSUB600:OPEN"CAS:"+NM$ FOR OUTPUT AS #1
6430 FOR I=1 TO NR:PRINT #1,TD$(I) :NEXTI:CLOSE #1
6440 CLS:LOCATE14,10:PRINT"FIN PROGRAMA":RETURN 4530
6500 REM --------
6501 REM CONFIRMA
6505 P1=1:P2=10
6510 A1$=MID$(A$,P1,P2)
6513 IF CP$="" THEN Z1=VAL(A1$):FZ=0:GOTO 6525 ELSE FZ=1
6516 IF LEN(A1$)>10THENA1$=LEFT$(A1$,10)ELSEA1$=A1$+SPACE$(10-LEN(A1$))
6520 IF LEN(CP$)>10THENCP$=LEFT$(CP$,10)ELSECP$=CP$+SPACE$(10-LEN(CP$))
6525 O=OP-5+(FZ*6)
6530 ON O GOSUB 6600,6610,6620,6630,6640,6650,6605,6615,6625,6635,6645,6655
6550 RETURN
6600 IF Z1>Z THEN FC=1 ELSE FC=0
6602 RETURN
6605 IFA1$>CP$THENFC=1ELSEFC=0
6607 RETURN
6610 IFZ1<ZTHENFC=1ELSEFC=0
6612 RETURN
6615 IFA1$<CP$THENFC=1ELSEFC=0
6617 RETURN
6620 IFZ1>=ZTHENFC=1ELSEFC=0
6622 RETURN
6625 IFA1$>=CP$THENFC=1ELSEFC=0
6627 RETURN
6630 IFZ1<=ZTHENFC=1ELSEFC=0
6632 RETURN
6635 IFA1$<=CP$THENFC=1ELSEFC=0
6637 RETURN
6640 IFZ1=ZTHENFC=1ELSEFC=0
6642 RETURN
6645 IFA1$=CP$THENFC=1ELSEFC=0
6647 RETURN
6650 IFZ1<>ZTHENFC=1ELSEFC=0
6652 RETURN
6655 IFA1$<>CP$THENFC=1ELSEFC=0
6657 RETURN
6800 REM --- BUSCA CAMPOS ---
6808 J=0
6810 FOR I=1 TO 60 STEP 6
6815 J=J+1
6817 FOR K=1 TO 10
6820 IF P7$(K)=B2$(I) THEN 6840
6830 NEXT K:TP(J)=0:GOTO 6850
6840 D$(J)=B2$(I):TP(J)=VAL(B2$(I+1)):LC(J)=VAL(B2$(I+2)):ND(J)=VAL(B2$(I+3)):DX(J)=VAL(B2$(I+4)
):CY(J)=VAL(B2$(I+5)):CX(J)=(DX(J)+11):DY(J)=CY(J)
6850 NEXT I
6860 RETURN
6900 REM --- BUSCA CAMPO ---
6908 J=0
6910 FOR I=1 TO 60 STEP 6
6915 J=J+1
6920 IFP7$(K)=B2$(I)THEN6940
6930 NEXTI:ER=3:RETURN20000
6940 RETURN
7000 REM ----------
7010 REM LECT.DATOS
7030 RESTORE 10080
7040 READ NV
7050 FOR I = 1 TO NV:READ NUC(I)
7060 NEXT I
7070 ON OPZ GOSUB 7200,7220,7230,7230
7071 NUC=NUC(OPZ):IFOPZ=2THENK1=1:K2=NUC-10:J=1:GOTO7074
7072 IFOPZ=1THENNUC=60:FORI=1TO20:D$(I)="":NEXTI:FORI=0TO9:READD$(I*6+1):NEXTI:GOTO7078
7074 FOR I=1 TO NUC:READ D$(I)
7076 NEXT I:IF OPZ=2 THEN 7082
7077 IF OPZ=3 THEN K1=1:K2=NUC-9:J=1:GOTO 7082
7078 FOR J=1 TO 10
7079 RESTORE10110
7080 K1=1+6*(J-1):K2=K1+5
7082 FOR I=K1 TO K2
7090 READ TP(I),LC(I),ND(I),DX(I),CX(I),CY(I):CY(I)=CY(I)+(J-1)
7100 DY(I)=CY(I)
7105 NEXTI
7110 IF JK=2 THEN RETURN
7111 IF OPZ>2 THEN 7130
7112 IF OPZ=2 THEN JK=2:FOR J=1 TO 5:RESTORE 10240:K1=J*2+1:K2=K1+1:GOSUB 7082:NEXT:JK=0:RETURN
7115 NEXT J
7120 GOTO 7180
7130 IF OPZ=3 THEN JK=2:FOR J=1 TO 5:RESTORE 10300:K1=J*2+2:K2=K1+1:GOSUB 7082:NEXT:JK=0
7180 RETURN
7200 RESTORE 10100:RETURN
7210 RESTORE 10160:RETURN
7220 RESTORE 10210:RETURN
7230 RESTORE 10270:RETURN
9000 REM
9030 RETURN
10000 REM -----
10010 REM DATOS
10025 DATA "+","-","*","/","^","> ","< ",">=","<=","= ","<>"
10030 DATA " EL ORDENADOR PARA EL ORDENADOR "
10040 DATA 4
10050 DATA "D E F I N I C I O N E S","I N S T R U C C I O N E S","L I S T A D O","E J E C U C I
O N"
10070 DATA 30,31,28,29,20,16,15,13,25,19,17,0,0,0,0
10080 DATA 3
10090 DATA 6,12,13
10100 DATA 1,2,3,4,5,6,7,8,9,10
10110 DATA 3,10,0,1,3,10
10120 DATA 1,1,0,16,16,10
10130 DATA 1,2,0,20,20,10
10132 DATA 1,2,0,25,25,10
10134 DATA 1,2,0,30,30,10
10136 DATA 1,2,0,35,35,10
10160 DATA "LEE",10,"ESC",10,"BOR",0,"BUS",3,"CAL",4,"DEC",4,"REP",10,"FIN",0
10180 DATA 1,8,1,7,25,11
```

```
10190 DATA 2,8,1,7,25,13
10210 DATA "NOMBRE FICHERO","LONGITUD", , , , , , , , , ,
10220 DATA 3,6,0,1,16,9
10230 DATA 1,2,0,24,35,9
10240 DATA 3,1,0,2,14,14
10250 DATA 3,1,0,20,32,14
10270 DATA "Num.","Instruccion","Parametro",1,2,3,4,5,6,7,8,9,10
10275 DATA 1,2,0,1,2,9
10280 DATA 3,10,0,10,10,9
10290 DATA 3,6,0,23,25,9
10300 DATA 3,10,0,2,5,13
10310 DATA 3,10,0,20,23,13
11000 REM POS CUR
11030 LOCATE X,Y:RETURN
12000 X=5:Y=8:GOSUB11000
12010 PRINT"Nombre    Tip Lon. Dec. Col. Lin.";
12020 GOTO2190
13000 REM --- AYUDA ---
13050 LOCATE 0,22
13055 PRINT "AYUDA (barra espaciadora para continuar)";
13060 GOSUB 13600
13110 PRINT "FLECHAS: mueven en las 4 direcciones    ":
13120 GOSUB 13600
13130 PRINT "RETURN: alinea y convalida introduccion ";
13140 GOSUB 13600
13150 PRINT "F1: vuelve al menu principal            ";
13160 GOSUB 13600
13170 PRINT "F2: anula toda la introduccion          ";
13180 GOSUB 13600
13190 PRINT "F3: borra un caracter                   ";
13200 GOSUB 13600
13210 PRINT "F4: permite la insercion de espacios    ";
13220 GOSUB 13600
13222 PRINT"F5: graba toda la introduccion           ";
13224 GOSUB 13600
13230 PRINT "                          CTRL Y (AYUDA) ";
13300 RETURN
13600 A$=INKEY$:IF A$= "" THEN 13600
13605 IF A$<> CHR$(32) THEN 13600
13610 LOCATE 0,22
13620 RETURN
15000 CLS:LOCATE 3,9:PRINT"ERROR DE LOGICA O CALCULO ERRONEO":LOCATE 2,12:PRINT"COMPROBAR Y REPE
TIR LA EJECUCION":LOCATE 10,15:PRINT"Pulsar una tecla"
15010 IF INKEY$="" THEN 15010 ELSE RESUME 230
20000 CLS:LOCATE6,10:PRINT"ERROR ";ER;" Pulsar una tecla"
20002 IF INKEY$="" THEN 20002 ELSE RETURN 230
21000 RETURN:REM --- STOP ---
```

RELACION DE LAS PRINCIPALES VARIABLES PHILIPS VG 8010

B$(*) = buffer data entry
B2$(*) = contiene la descripción, el tipo, la longitud, la posición, etc., de los campos definidos por el usuario en la fase Definiciones
D$(*) = descripciones campos de la data entry
TP(*) = tipo de campos de la data entry
LC(*) = longitud de los campos de la data entry
ND(*) = número decimales campos de la data entry
DX(*), **DY(*)** = coordenadas de la descripción campos de la data entry
CX(*), **CY(*)** = coordenadas campos de la data entry
TF(*) = contiene los códigos de teclas de función
M3 = contiene el número máximo de instrucciones
OP$(*) = contiene los 11 operadores matemáticos y lógicos previstos
IST$(*) = contiene las instrucciones
NT(*) = contiene los números de las instrucciones
PER$(*) = contiene el periférico o el parámetro de las instrucciones
PAR$ (*, *) = contiene los campos, los operadores o las constantes a que se refieren las instrucciones
FR$(*) = contiene las 8 instrucciones previstas
NP(*) = contiene el número máximo de parámetros previstos para cada instrucción
S0(*) = contiene la posición de las instrucciones en FR$(*)
PROG$ = contiene el nombre del programa
NV = contiene el número de voces del menú
VO$ = contiene las voces del menú
FS = indica el número (de 1 a 15) de la tecla función apenas pulsada
NM$ = contiene el nombre del fichero definido en la fase Definiciones 2
NR = contiene el número de registros del fichero definido en la fase Definiciones 2
LR = contiene la longitud de cada registro del fichero definido en la fase Definiciones 2
SK$ = contiene la "S" para los campos que forman parte del fichero definido
G5 = contiene el número de instrucciones introducidas + 1
OPZ = contiene el número de referencia para la presentación máscara y la lectura de las DATA
NUC = contiene el número de campos de la data entry
IST$ = contiene la instrucción que se ejecuta en aquel momento
NT = contiene el número de la instrucción que se ejecuta en aquel momento
PER$ = contiene el periférico o parámetro de la instrucción ejecutada en aquel momento
P7$(*) = contiene los campos, los operadores, etc., de la instrucción ejecutada en aquel momento
Z,Z1, **Z2,Z3** = se utilizan para cálculos y comparaciones numéricas
A1$**, **CP$ = se utilizan para comparaciones alfanuméricas
FC = contiene el resultado de la comparación (1 o 0)
G9, B7 = se utilizan como switches

Versión disco

TD$ = variable de comunicación con disco
WQ$ = contiene el número de registros que hay en el disco
RE = número del último registro leído
R8 = número del último registro escrito

Versión cinta

TD$(*) = contiene los registros leídos de cinta o de grabación en cinta
RE = número del último registro leído
R8 = número del último registro escrito
H6 = se utiliza como switch

Actualización. La actualización consiste en introducir, para un determinado código, el importe movido (en el campo Movimiento) y en calcular el nuevo valor del SALDO, con la expresión:

SALDO = SALDO − MOVIMIENTO

El significado de esta expresión es: el nuevo saldo es igual al anterior menos el movimiento. Y, aunque lógicamente nada impide realizarlo de la manera indicada, conviene adoptar otro método de cálculo introduciendo una variable auxiliar: el RESTO. De esta manera, se tiene:

RESTO = SALDO − MOVIMIENTO

que expresa el nuevo valor del saldo, a memorizar con la expresión:

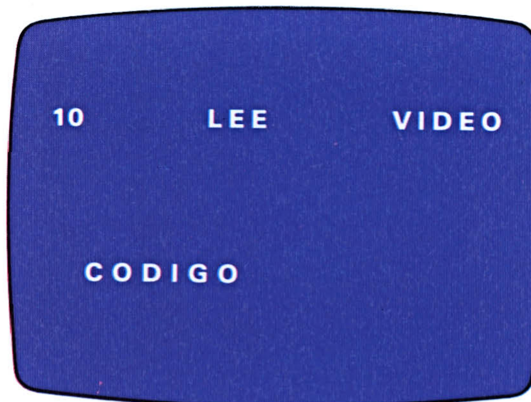
SALDO = RESTO + 0 (el operador + y el segundo sumando derivan de la sintaxis de la instrucción)

Por tanto, el programa puede desarrollarse con los siguientes pasos:

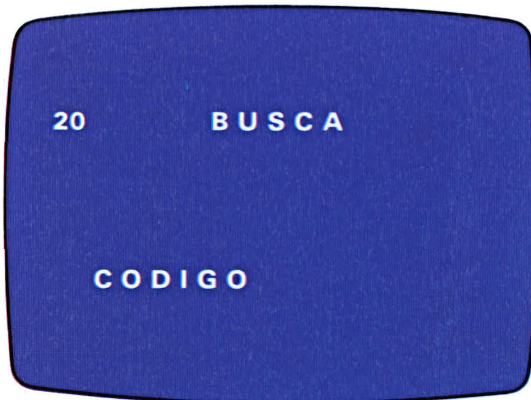
— Introducción del CODIGO al que se refiere el movimiento (línea 10)
— Búsqueda en el fichero (línea 20)
— Representación, para comprobación, de la máscara completa (30)
— Introducción del MOVIMIENTO (40)
— Cálculo del RESTO (50)
— Cálculo del SALDO (60)
— Representación de la máscara (70)
— Transferencia de los nuevos valores al correspondiente fichero (80)

Sin embargo, en la forma presentada, el programa tiene un defecto. Si en la instrucción 20 no se encuentra el código, el proceso continúa a pesar de ello, por lo que el desarrollo de las instrucciones que siguen deberá condicionarse (con la instrucción DECIDE). En este caso particular, el error no puede producirse, porque el fichero se ha creado con una longitud igual al código máximo utilizable (9).
Téngase en cuenta que al escribir este segundo programa no es necesario introducir de nuevo las definiciones, porque pueden volverse a utilizar las del programa anterior. En este caso, el segundo programa debe escribirse sin apagar el ordenador para no perder las definiciones anteriores.

Introducción del código para la selección del registro. El valor introducido por pantalla, según el formato especificado en la fase de definición, se transfiere al campo de nombre CODIGO.



Busca el valor contenido en el campo CODIGO y se compara con los que hay en el fichero, en los bytes reservados para este campo.



Presentación en pantalla del contenido del registro, o sea del importe inicial y del último saldo, correspondiente al movimiento anterior.

30 ESCRIBE VIDEO

CODIGO DESCR
IMPORTE SALDO

Lectura del nuevo movimiento.

40 LEE VIDEO

MOVIMIENTO

Cálculo del resto, o sea del nuevo valor del saldo.

50 CALCULA

RESTO SALDO
– MOVIMIENTO

Transferencia al campo saldo.

60 CALCULA

SALDO RESTO
+ 0

Presentación en pantalla de los nuevos valores.

```
70      ESCRIBE      VIDEO

   CODIGO          DESCR
   IMPORTE         SALDO
```

Escritura en disco.

```
80      ESCRIBE      DISCO

   CODIGO          DESCR
   IMPORTE         SALDO
```

Fin de tarea.

```
99        FIN
```

EJEMPLO DE SALIDA EN IMPRESORA (C-64)

```
10      LEE         V
                            CODIGO
                            DESCR
                            IMPORTE

20      CALCULA
                            SALDO
                            IMPORTE
                            +
                            0

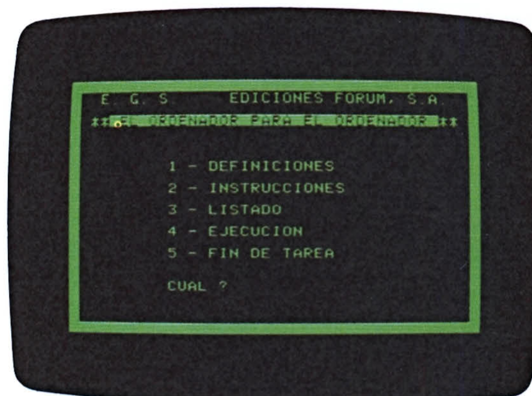
30      ESCRIBE     V
                            CODIGO
                            DESCR
                            IMPORTE
                            SALDO
```

```
40      ESCRIBE     D
                            CODIGO
                            DESCR
                            IMPORTE
                            SALDO

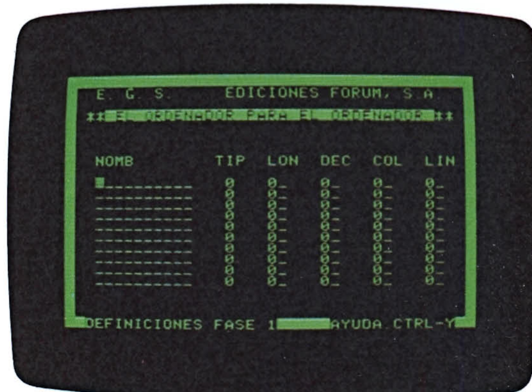
60      REPITE      9
                            10
                            20
                            30
                            40
                            FIN

99      FIN
```

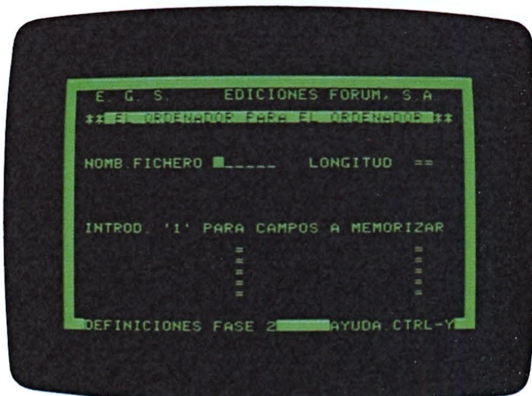
Menú principal. Este procedimiento no utiliza los gráficos, por lo que la versión Apple es formal y funcionalmente idéntica a las otras.



Introducción de los campos. En esta fase deben definirse los atributos de todos los campos que se van a utilizar a continuación.



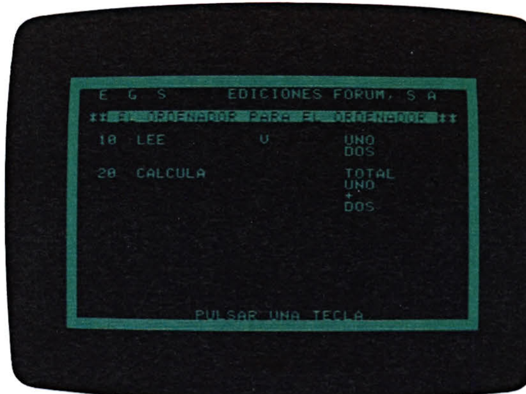
También en esta versión, al final de la fase de definición hay que especificar entre todos los campos previstos los que se desea memorizar en el fichero.



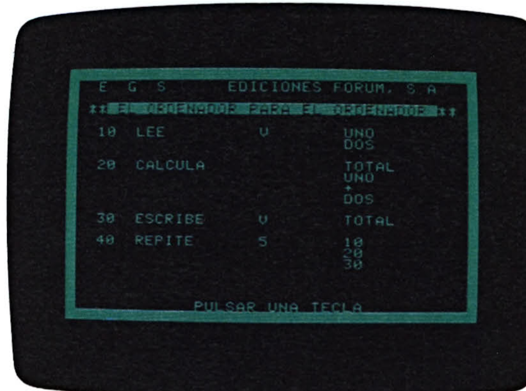
La foto presenta la máscara vídeo correspondiente a la introducción de las instrucciones. Su estructura y su utilización son idénticas a las de las otras versiones.



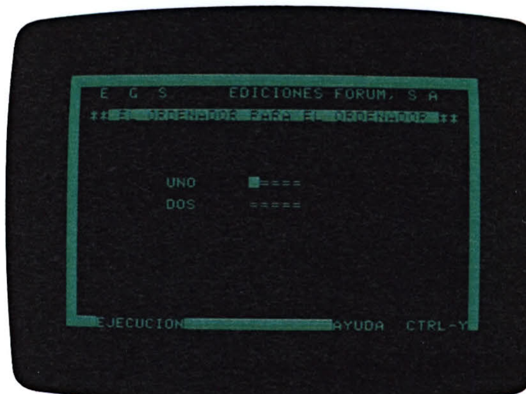
Ejemplo de programa. Los dos líneas, la 10 y la 20, se emplean para leer dos valores (respectivamente memorizados en las variables UNO y DOS) y para calcular la suma.



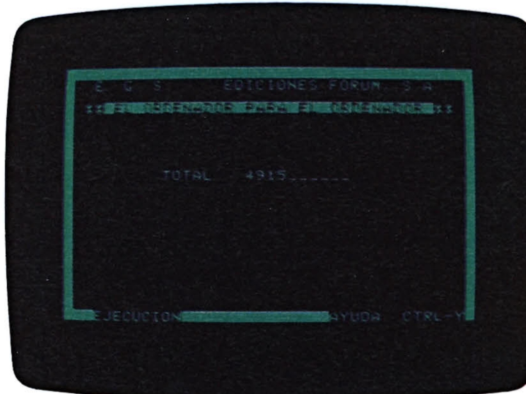
El completado del programa anterior se realiza con dos instrucciones: la número 30 para la escritura en pantalla del resultado y la número 40 para la activación del bucle de repetición.



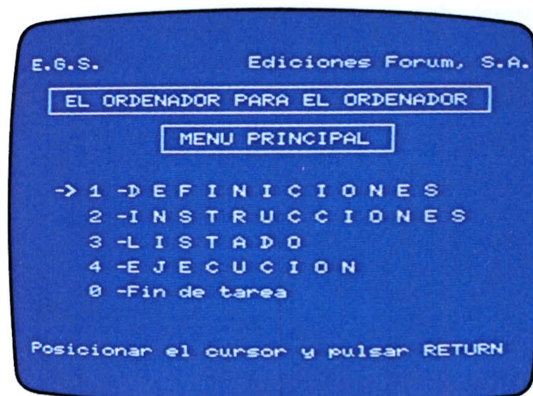
Esta máscara vídeo aparece en el momento de la ejecución del programa y corresponde a la línea número 10. Naturalmente, todos los atributos de los campos deben haber sido definidos previamente.



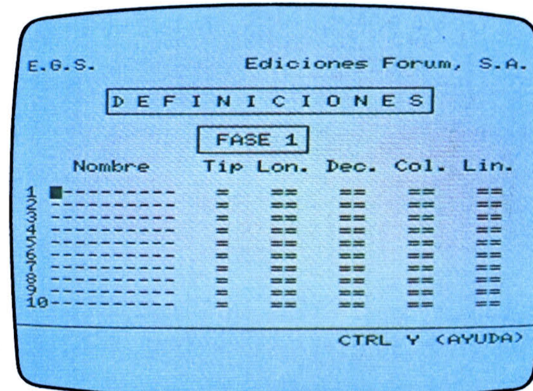
La foto muestra el efecto de la línea 30, que contiene la instrucción ESCRIBE (en vídeo). El valor numérico que aparece al lado del nombre del campo TOTAL es el resultado del proceso sobre los valores de los dos campos anteriores.



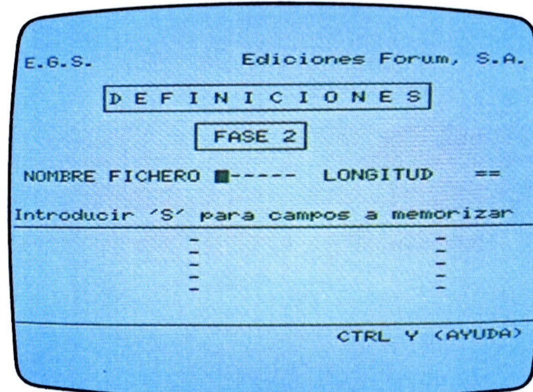
Menú principal. También para la versión MSX; aparte de diferencias marginales, resulta válido todo lo dicho para las versiones anteriores.



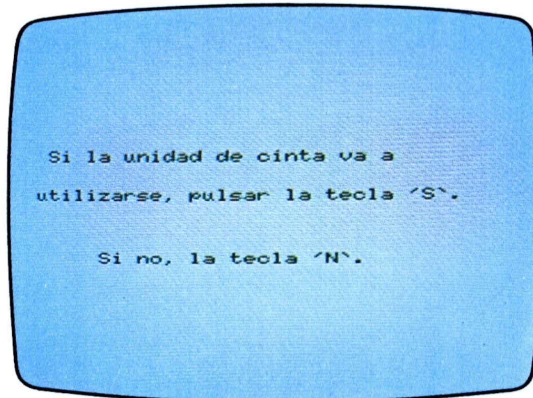
Definiciones. Comparando esta versión con las otras resulta evidente una coincidencia casi absoluta.



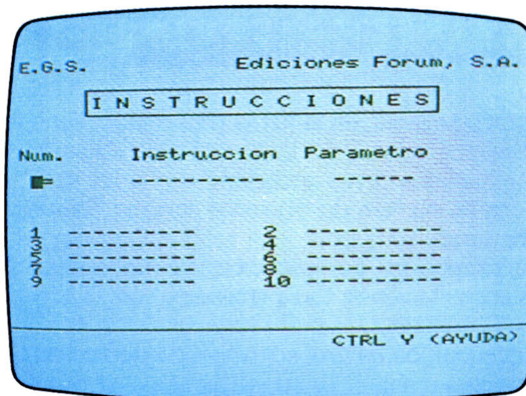
Al final de los atributos de los campos hay que definir cuáles, entre todos los introducidos, deberán constituir el registro del fichero.



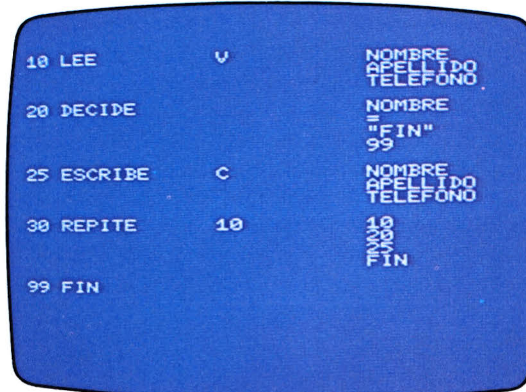
En esta versión existe la diferencia introducida por el uso de la unidad de cinta en lugar de la de disco.



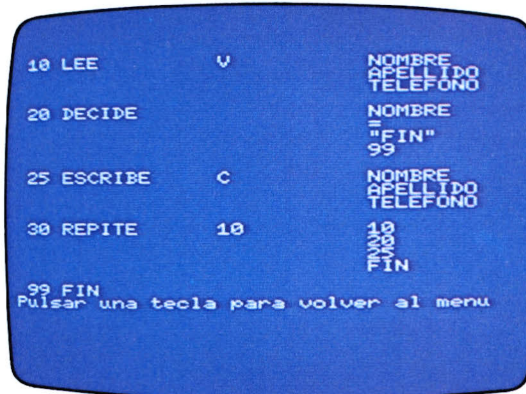
Formato de las instrucciones. La máscara vídeo también es similar a las anteriores. Obsérvese que no se ha previsto ningún control sobre el periférico utilizado, por lo que debe ser el usuario quien no ha de llamar la unidad de disco (si su sistema no la prevé).



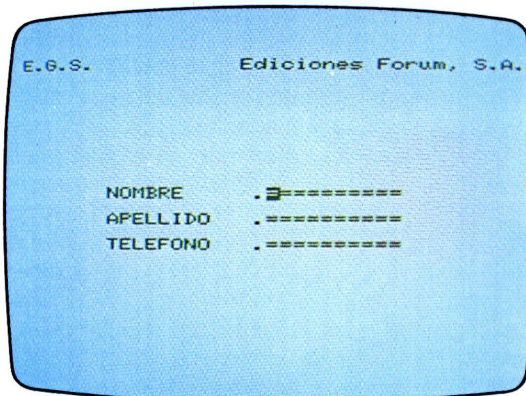
Ejemplo de programa. Las funciones realizadas son:
línea 10: lee de pantalla los campos especificados (por lo que se trata de una data entry)
línea 20: comprueba si en el campo NOMBRE se ha introducido el valor FIN. En caso afirmativo, el programa termina
línea 25: transfiere la introducción a la cinta (periférico C = cinta)
línea 30: activa el bucle para diez veces.
En este programa se muestran las dos maneras de salir de una introducción repetitiva al final del bucle (después de diez repeticiones) o al cumplirse una determinada condición (línea 20).



La foto muestra el desarrollo de un listado del programa. Al final del mismo, el procedimiento se pone en espera de una tecla cualquiera antes de proseguir.



Máscara vídeo generada por la instrucción número 10.



Extensiones y modificaciones

El procedimiento, en la forma presentada, tiene finalidades didácticas, y por tanto está orientado a usuarios poco expertos cuyas necesidades pueden satisfacerse con un número de funciones limitado. Para aplicaciones más complejas, con programas para problemas específicos de aplicación, son necesarias algunas implantaciones. Las principales son:

- Inserción de un procedimiento de gestión en la impresora
- Inserción de instrucciones análogas a las llamadas de subrutinas
- Posibilidad de memorizar el programa

La primera, gestión de la impresora, no presenta dificultades de concepto, reduciéndose a una subrutina de impresión parametrizada. Una forma muy sencilla puede obtenerse insertando en los códigos de los periféricos conocidos la sigla IM (IMPRESORA) y una subrutina que envíe a impresión los contenidos de los campos especificados en la instrucción. Esta subrutina puede obtenerse modificando la instrucción ESCRIBE correspondiente al vídeo.
El segundo punto es mucho más complejo, y la ilustración debe empezar especificando las motivaciones. En la escritura de un programa es frecuente utilizar instrucciones bajo condición. Por ejemplo, en este macrolenguaje, la instrucción DECIDE envía a ejecución otras instrucciones al producirse una determinada condición. Sin embargo, para un funcionamiento correcto, es necesario que dichas instrucciones no se realicen en el flujo normal.
Analizando la lógica de este procedimiento, es evidente que las condiciones expuestas no se producen. Cada instrucción está identificada por un número de línea y se envía de alguna manera a ejecución. Es decir, si por ejemplo se quiere emitir un mensaje por pantalla al producirse una cierta condición, la instrucción a utilizar es DECIDE... "N", con N número de línea a ejecutar si el test (DECIDE) da un resultado positivo. En cualquier caso, este número de línea (N) debe estar presente y, por tanto, en la versión actual, la función correspondiente debe ejecutarse dos veces: una en el flujo normal y otra al cumplirse la condición. En consecuencia hay que insertar una lógica que en el flujo principal impida la ejecución de determinadas instrucciones, las que sólo deben activarse al producirse una condición. Una solución consiste en prever números de línea de 3 cifras. De esta manera, las instrucciones a activar bajo condición pueden numerarse con valores superiores a 99 y, por tanto, no se ejecutan en el flujo normal, que tiene este valor como límite del desarrollo.
Procediendo así se tiene una forma extremadamente estructurada: las instrucciones hasta 99 constituyen el flujo principal, ejecutado en orden creciente, mientras que las numeradas a partir de 100 constituyen procedimientos «aislados», activados solamente si son llamados desde el flujo principal.
Sin embargo, existe una vía indirecta para activar esta lógica, que consiste en un uso particular del procedimiento: las instrucciones a ejecutar sólo bajo condición deben numerarse de manera que ocupen la parte final, inmediatamente antes de la línea 99, y precedidas por una instrucción de salto (DECIDE) que «desvíe» el flujo principal a la 99 sin interesar las líneas intermedias. Naturalmente, esta instrucción DECIDE debe contener un test que proporcione siempre un resultado positivo; de esta manera, la instrucción se hace análoga a la GOTO. Sin embargo, a pesar de que esta solución proporciona el resultado deseado, no es la mejor, puesto que reduce notablemente la estructuración de los programas realizados con este macrolenguaje.
La última implantación sugerida, o sea la posibilidad de memorizar el programa, es la más importante, incluso por las particulares implicaciones que comporta su realización.
El programa está contenido en 4 variables dimensionadas (de las que una bidimensional contiene los parámetros, el primer índice identifica la instrucción y el segundo los correspondientes parámetros) y para memorizarlo basta con transferir secuencialmente al disco (o a la cinta) cada matriz.
Análogamente, para volverlo a cargar en la memoria basta prever una rutina de lectura que extraiga el contenido del fichero subdividiéndolo en cada una de las 4 variables, que son:

NT(*)	: Contiene los números de línea
IST$(*)	: Instrucciones
PER$(*)	: Parámetros (periféricos)
PAR$(*,*)	: Campos

Por tanto, la temporización del programa consiste en un bucle entre 1 y el número máximo de las instrucciones (o sea el dimensionado de cada matriz) que sume el contenido de cada matriz en el orden visto. Por ejemplo, el registro genérico "I" está compuesto por:

A$ = STR$(NT(I)) + IST$(I) + PER$(I) + A1$

donde A1$ es la suma de todos los parámetros previstos en la instrucción:

A1$ = PAR$(I,1) + PAR$(I,2) + ... + PAR$(I,10)

Naturalmente, el cálculo indicado en A1$ es indicativo, porque en el programa deberá ser un bucle.

Programas para la profesión



INTRODUCCION

Esta última parte se dedica a programas de tipo «profesional», o sea orientados a usuarios que deseen estructurar las posibilidades que ofrece un ordenador en el ámbito de su propio trabajo.
Sin embargo, hay que establecer una premisa. Todos los procedimientos ilustrados en esta obra están confinados en la categoría de «ordenadores domésticos», o sea máquinas de un costo limitado y empleadas normalmente como videojuego.
La propuesta de implantar en este tipo de máquinas un software para empleos profesionales puede parecer pretenciosa e inadecuada, pero en realidad no existe ningún impedimento para ello, puesto que los ordenadores domésticos de la nueva generación tienen muy poco que envidiar a sus hermanos mayores. Si comparamos atentamente las características de un ordenador doméstico de buena calidad con las de uno personal, las diferencias pueden ser muy marginales. Las características sobre las que basar un juicio son:

- capacidad y tiempo de acceso de las memorias de masa (diskettes)
- software que la máquina puede albergar
- capacidad de memoria (muy marginalmente)

En particular, el último punto, capacidad de memoria, es uno de los parámetros que más inducen a evaluaciones erróneas, porque los intérpretes Basic o los compiladores tanto Basic como de otros lenguajes no direccionan más allá de 64 K de memoria, por lo que en la mayor parte de las aplicaciones es del todo inútil disponer de áreas mucho más amplias.
Incluso los ordenadores domésticos modernos (como los MSX) han superado con mucho este límite. También en lo que respecta a la gestión de la memoria de masa, no existe una diferencia sustancial entre los dos tipos de máquinas, e incluso sucede que algunos ordenadores domésticos tienen unidades de disco de 500 Kbytes en diskettes de 3" 1/2, contra los 360 Kbytes en 5" 1/4 de los personales. Por tanto, la única diferencia verdadera reside en el software.
Mientras que para la categoría PC se han desarrollado desde hace tiempo sistemas operativos avanzados y programas de empleo generalizado, para los ordenadores domésticos, el esfuerzo de los productores de software se ha orientado principalmente a los videojuegos, y sólo recientemente ha empezado la difusión de software «profesional».
Los cuatro campos que siguen

- el tablero electrónico
- la Base de Datos
- los gráficos de ordenador
- el tratamiento de textos

constituyen un ejemplo de cómo es posible automatizar algunas funciones en el ámbito profesional, incluso con un ordenador doméstico.
En cualquier actividad profesional hay algunos aspectos comunes como:

- proceso de valores numéricos
- gestión de archivos
- preparación de gráficos para representar
- preparación de textos (cartas, listas, etc.).

Todos estos puntos tienen las mismas características, independientemente del sector en cuyo ámbito se desarrollan:

1 / Conservan la misma lógica
2 / Necesitan un elevado período de generalización
3 / Si se desarrollan con un ordenador, no deben requerir ninguna otra preparación específica.

A partir de esta observación se han desarrollado programas que responden a las características enunciadas y que realizan las funciones vistas. Para los procesos de valores numéricos se ha producido el tablero electrónico, para los archivos la Base de Datos, los gráficos se han desarrollado en el tercer paquete y para la preparación de los textos se ha previsto el programa de tratamiento de textos.
Utilizando adecuadamente el «paquete» de software así constituido, es posible delegar a las máquinas muchas de las operaciones, no creativas, que se efectúan en la actividad laboral. Los cuatro programas se presentan, como de costumbre, bajo el doble aspecto de utilización y de desarrollo.
La parte dedicada a la utilización está orientada a quienes desean aprovechar el programa sin entrar en los detalles de las distintas funciones.
El desarrollo muestra en detalle la lógica seguida y coloca al usuario en posición de aportar todas las modificaciones y las implantaciones que considere necesarias para adaptar mejor el software a sus necesidades específicas.
Téngase en cuenta que esta sugerencia no es una contradicción por la que se invita a «personalizar» un programa generalizado, sino que se debe a una motivación muy precisa. Por la extremada generalidad que deben tener, estos programas en algunos casos pueden resultar farragosos, poco adaptados y excesivamente lentos. Si se tiene la posibilidad de modificarlos, pueden eliminarse las partes no interesantes, obteniendo versiones más limitadas pero más «ágiles» para su aplicación específica. Naturalmente, una vez realizadas las modificaciones habrá que recompilar todo el procedimiento.

El tablero electrónico

El tablero electrónico está entre los más útiles, y sin duda es el más versátil de todos los programas de empleo generalizado. Su lógica se basa en el hecho de que cualquier proceso de tipo numérico puede representarse en una tabla cuyo contenido puede estar constituido bien por valores sobre los cuales efectuar el proceso, bien por las fórmulas a aplicar. Esquematizado de esta manera el problema, es posible realizar un programa que puede adaptarse a la mayor parte de las necesidades de cálculo.
La versión presentada permite realizar cálculos con fórmulas definidas por el usuario o con fórmulas «internas», o sea con algoritmos predefinidos y dedicados al desarrollo de procesos que se realizan con frecuencia (por ejemplo, el sumatorio o la media de valores, etcétera). El campo de aplicación es prácticamente ilimitado, desde el uso como sencilla calculadora, aunque muy potente, hasta el desarrollo de cálculos estadísticos o de funciones matemáticas: por tanto, su empleo también puede ser ventajoso en el ámbito docente en las escuelas.
Un ejemplo de sus posibilidades es en la distribución de un gasto comunitario; una vez memorizada la tabla milesimal, ya no es necesario realizar ningún cálculo, puesto que el tablero electrónico dará los resultados. La intervención del usuario estará limitada a la introducción de los importes a repartir. Con la misma facilidad se puede crear una aplicación para la gestión de un almacén (entradas, salidas, existencias), o para el análisis de la rentabilidad de una inversión.

Uso del programa

Con «tablero electrónico» se indica una «familia» de programas de empleo generalizado y particularmente adecuados para la realización de cálculos.
Su empleo prescinde del conocimiento de cualquier lenguaje de programación, por lo que estos programas deben contener un intérprete que traduzca las instrucciones de alto nivel (impartidas por el usuario) a códigos máquina o que las «explote» en instrucciones más elementales pertenecientes a un lenguaje de programación.
El mejor enfoque para la preparación de un software de este tipo es el lenguaje Assembler, puesto que permite interpretar directamente las instrucciones y aprovechar completamente las posibilidades de la máquina. Sin embargo, el uso de este lenguaje requiere conocimientos no corrientes e implica una notable dedicación. Como alternativa puede utilizarse un lenguaje de programación más evolucionado, como el Basic, obteniendo una mayor facilidad en la escritura del programa, si bien con algunas limitaciones, en particular una menor velocidad de ejecución y un aprovechamiento incompleto de la memoria disponible.
El programa presentado se ha estructurado utilizando el lenguaje Basic, y por tanto está sujeto a las limitaciones indicadas. En compensación ofrece la posibilidad de modificaciones e implantaciones. Naturalmente, para su empleo hace falta la versión compilada, puesto que, de otro modo, los problemas de tiempo serían demasiado grandes para que fuese utilizable.
Como ya se ha comentado, un tablero electrónico se emplea para el desarrollo de cálculos. En sustancia, para obtener este resultado es necesario prever las siguientes funciones principales:

- Introducción
- Desarrollo del cálculo
- Presentación de los resultados.

La diferencia sustancial con un programa análogo escrito en Basic y no orientado a un empleo generalizado reside en el modo de tratar las variables y en la presentación del resultado.
Por ejemplo, supongamos que se quiere escribir un programa que permita la realización de cualquier operación entre dos datos.
En Basic, las funciones a realizar son:

1 / Lectura por teclado de dos valores
2 / Lectura por teclado de la operación a realizar, o sea de los símbolos (+, –, *, /) correspondientes
3 / Realización del cálculo
4 / Presentación del resultado.

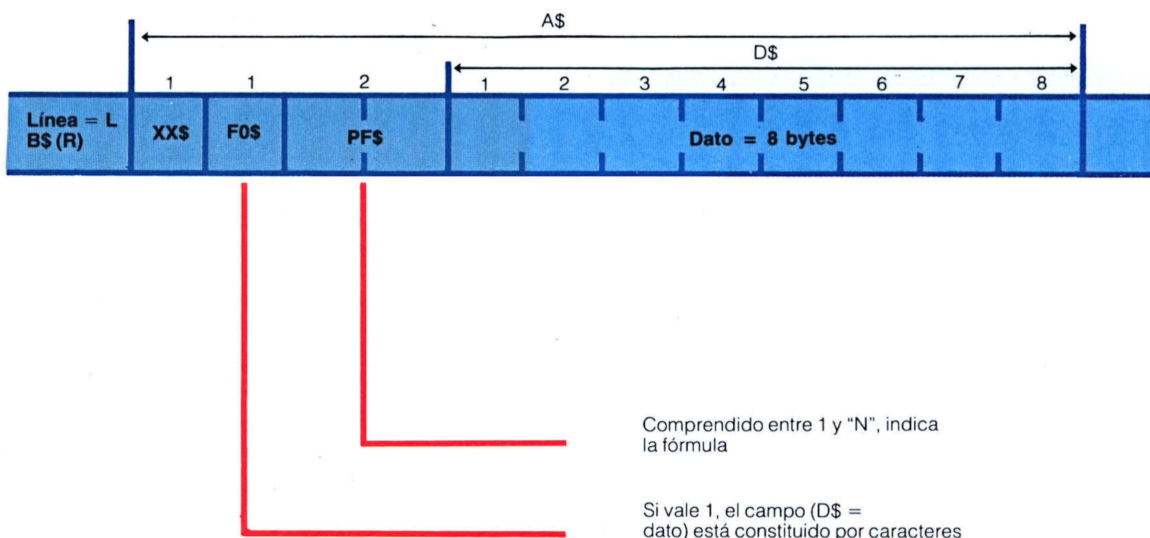
La primera función consiste en asignar dos nombres simbólicos a las dos variables y en adquirir su valor por teclado, por ejemplo con una instrucción INPUT.
La segunda función necesita una lectura más que tiene el significado de código, en base al cual se realizará el cálculo. El desarrollo del cálculo es el punto difícil. Si se conociese *a priori* el tipo de operación no habría dificultades. En cambio, si se desea parametrizar, hay que prever todas las posibles operaciones, escribiendo una subrutina para cada una y después utilizar la correspondiente al símbolo introducido. Por tanto, se necesita una matriz de control en la que memorizar (con una DATA), una vez introducidos, los símbolos previstos y, después determinar con un bucle cuál se

ha pedido y, en base a la respuesta, llamar la correspondiente rutina.

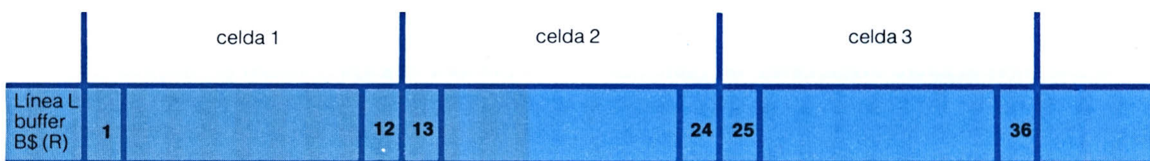
En este ejemplo, con sólo dos variables, no hay problemas de una cierta complejidad. Sin embargo, deben tomarse algunas precauciones: como mínimo es necesario que el programa compruebe la correcta sucesión de la introducción de los valores. Por ejemplo, suponiendo que las dos variables tengan los nombres simbólicos A y B y que la subrutina que ejecuta el cociente prevea el cálculo de A/B, es necesario que la introducción sea controlada para asegurar la correcta asignación de los valores. Es decir, el usuario debe estar informado de cómo se realiza el cálculo y adónde (en qué variables) se transfieren los valores numéricos que intervienen en el cálculo. Para resolver estos problemas, el mejor modo es parametrizar incluso cada operación simple, realizándola sobre dos variables diferentes a las de introducción. De esta manera, el usuario deberá indicar las variables a utilizar, evitando así un posible equívoco.

Las variables A y B utilizadas para la introducción de los valores numéricos toman el significado de «dirección» en la que la rutina de cálculo debe extraer los valores. Por ejemplo, considérese una aplicación más generalizada que prevea el uso de 5 variables distintas de introducción y el desarrollo de un cálculo, en este caso, el cociente entre 2 de ellas. Las funciones que deben realizarse son:

ESTRUCTURA DE UNA CELDA



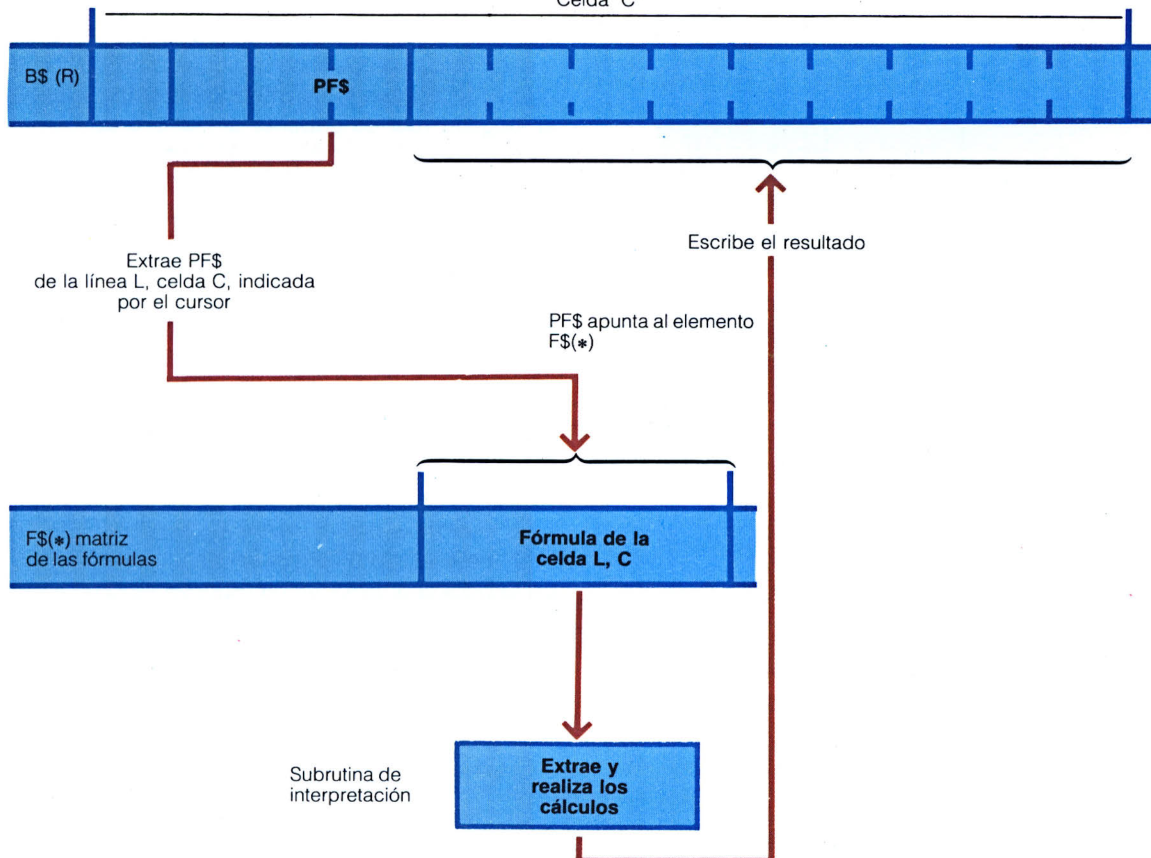
ESTRUCTURA DE UNA LINEA



Posición de principio celda "N" = BI = 1 + 12 * (N – 1)
Contenido celda "N" de la línea "L"

Contenido = MID$(B$(R),BI,12)

La estructura mostrada sólo es un ejemplo. En los listados, la longitud total de cada celda está limitada, por motivos de espacio, a 9 bytes: 7 para el valor, 1 para el puntero a las fórmulas (que por tanto son un máximo de 9) y 1 para el tipo



1 / Presentación de las 5 variables para la introducción de los valores
2 / Introducción de la indicación de cálculo.

La primera puede resolverse con una data entry que presente los nombres de las variables esperando una respuesta. La segunda debe estar constituida por una cadena que expresa el cálculo a realizar.
Terminada la fase de introducción tanto de los valores como de las operaciones, hay que activar una subrutina que analice la cadena, reconozca las variables contenidas en ella y las sustituya por los correspondientes valores numéricos.
Implantando la subrutina con todas las operaciones, se obtiene un módulo completamente parametrizado para el desarrollo de los cálculos, aunque en él aún subsisten limitaciones: la necesidad de volver a definir *a priori* los nombres de las variables y la asignación de los valores con una máscara vídeo que prevea su presentación.
Sin embargo, teniendo en cuenta que estos nombres simbólicos han perdido el significado de variables para adquirir el de direcciones de las que tomar los valores numéricos, es inmediato sustituirlos por un mecanismo de «selección posicional». El programa conserva los nombres simbólicos, mientras que el usuario ya no debe utilizarlos: en su lugar indica la posición.
Por ejemplo, en el caso de 5 variables, en la pantalla aparecerán 5 zonas, indicadas con las letras A, B, C, D y E, cada una de introducción. El usuario introducirá en las distintas zonas los respectivos valores numéricos y podrá indicar el cálculo especificando las dos zonas a considerar.
Desde este punto de vista, el desarrollo es completamente análogo al anterior. La única diferencia consiste en tener limitados los nombres de las variables a una sola letra. En realidad, esta metodología no permite utilizar variables internas.
Como se ha indicado, la posición de los 5 campos de introducción es fija y bien definida, por lo que puede utilizarse el posicionado del cursor para determinar cuál es la variable que está en fase de introducción.

Por ejemplo, suponiendo que cada campo tenga una longitud de 6 caracteres y que los 5 campos sean consecutivos, se tiene la situación:

A : empieza en la columna 1
B : empieza en la columna 7
C : empieza en la columna 13
D : empieza en la columna 19
E : empieza en la columna 25

Por tanto, basta con determinar la posición del cursor para saber a cuál de las 5 zonas corresponde el valor numérico que se está introduciendo.
Además, como la subrutina de cálculo no utiliza los mismos símbolos que la de introducción, no es necesario definir 5 variables diferentes, sino que basta con considerar un solo buffer de ubicación dividido lógicamente en 5 zonas. De esta manera puede preverse una cadena de 30 caracteres de longitud en la que realizar, con paso 6, las introducciones.
Para el desarrollo de los cálculos es suficiente tomar las partes interesadas (indicadas por la posición del cursor) y transferirlas, como parámetros, a la subrutina de cálculo. Cada zona en que está dividida la cadena de introducción recibe el nombre de «celda», y si el cursor está por ejemplo en la columna 19, se toma la celda número 4, marcada con la letra D, y viceversa, empleando la letra D, que ocupa la cuarta posición en el alfabeto, se selecciona la celda correspondiente que empieza en la posición 19 en el buffer de línea. Utilizando de manera análoga diferentes buffers, uno para cada línea de la pantalla, se dispone de una «retícula» de celdas en la que cada una está identificada por una línea y una columna.
En los ejemplos anteriores no se ha definido el cálculo del resultado de cada operación. Esto se ha hecho voluntariamente, puesto que está estrechamente ligado a la estructura final. Cada celda puede contener tanto un valor (numérico o alfanumérico) como una fórmula. En este caso no se presenta el verdadero contenido, como en los otros casos, sino el resultado de la fórmula.

Para memorizar la fórmula hay que utilizar una matriz diferente, porque la celda debe contener el resultado y, por tanto, a ésta hay que relacionarle un puntero que defina la cadena que representa el cálculo asociado a la celda. Además de este puntero hay que prever la indicación del tipo de contenido. El valor memorizado en una celda puede ser numérico o alfanumérico (caracteres). Para distinguir un caso de otro se utiliza un flag que forma parte de la celda y, por tanto, la longitud real es superior al número de caracteres presentados. En la página 2410 se han representado las escrituras de una celda y de una línea. El espacio, en bytes, reservado a la celda está dividido en dos partes, la primera dedicada a los punteros, y la otra al valor. Los punteros utilizados en esta versión son:

PF$ = indica la fórmula asociada a la celda
F0$ = tipo del dato ("1" = alfanumérico).

En la parte inferior de la figura se ha representado un ejemplo de extracción del contenido de una celda del buffer general de línea B$(R). El número del buffer, o sea el valor de R, está determinado por la línea en que se encuentra el cursor, mientras que el número de celda N está ligado a las columnas.
En la página anterior se ha representado un esquema de la lógica seguida en el desarrollo de las fórmulas asociadas a una celda (presentación del resultado).
El programa extrae de la matriz B$(R) la parte que corresponde a la celda examinada utilizando el método del gráfico de la página 2410. De esta cadena (A$) se toma el punto PF$ que, si es diferente de cero, indica la fórmula asociada a la celda. En este caso, PF$ (transformado en numérico) se utiliza para tomar de la matriz F$(*), que contiene la fórmula, la perteneciente a la celda examinada. La fórmula así obtenida se aplica, y el resultado se deposita en la adecuada zona de la celda.
La lógica descrita sólo es válida a nivel funcional, ya que en realidad, la subrutina que ejecuta las funciones descritas es muy compleja, sobre todo en la parte que se ocupa del desarrollo de los cálculos.

El área de trabajo del tablero electrónico está constituida por una retícula de 15 × 26 casillas de introducción. En la pantalla puede presentarse simultáneamente un máximo de 9 × 5.
La indicación de la celda considerada es proporcionada por la presentación de la misma en caracteres invertidos. En la versión C-64, los comandos para el desplazamiento son:
CRSR UP (↑) : mueve arriba
CRSR DOWN (↓) : mueve abajo
CRSR RIGHT (→) : mueve a derecha
CRSR LEFT (←) : mueve a izquierda



Desplazamiento de la pantalla. Posicionándose en los extremos de la pantalla, se produce el "scroll" del contenido de las celdas de todo el tablero, con la consiguiente actualización de los ejes de referencia (línea, columna). En este caso se ha activado un desplazamiento a la derecha: esto ha producido el "scroll" del contenido de la pantalla hacia la izquierda y la actualización del eje horizontal.



Al pedir un desplazamiento hacia abajo, se activa un procedimiento análogo al anterior. El contenido de la pantalla corre hacia arriba y el eje vertical se actualiza con cada desplazamiento. Esta operación permite presentar, en grupos de 9, todas las 15 líneas previstas en el tablero (de la letra A a la O).

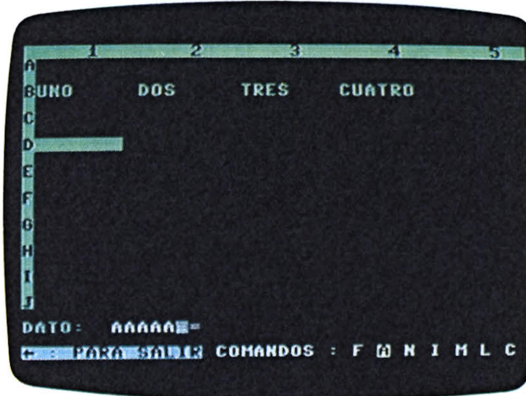


Introducción de datos. La introducción de los datos se produce en la zona inferior de la pantalla. El tipo de dato se selecciona con el comando activado (carácter amarillo de abajo a la derecha). Los comandos de introducción son:

F: para la introducción de fórmulas (máx. 30 caracteres)

A: para la introducción de datos alfanuméricos (máx. 7 caracteres)

N: para la introducción de datos exclusivamente numéricos (máx. 7 caracteres).



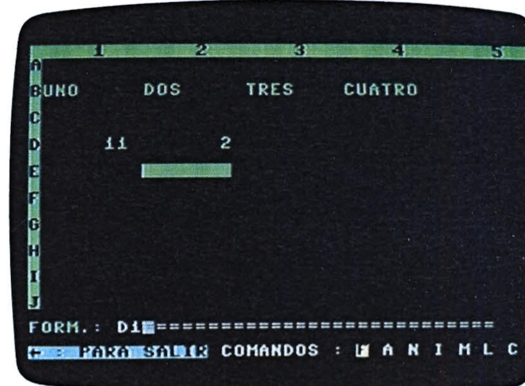
Una vez completada la escritura, mediante la tecla RETURN se confirma la introducción: así, el dato se transfiere a la celda en la que se había pedido el comando de introducción. En este caso, la cadena "AAAAA" se ha introducido en la celda D1.



Introducción de datos numéricos: La introducción de datos numéricos se realiza de manera análoga a la anterior: cualquier introducción diferente de un número será ignorada, excepto el signo "–". El dato numérico introducido se transfiere a la celda alineado a la derecha, como se ve en la celda D1.

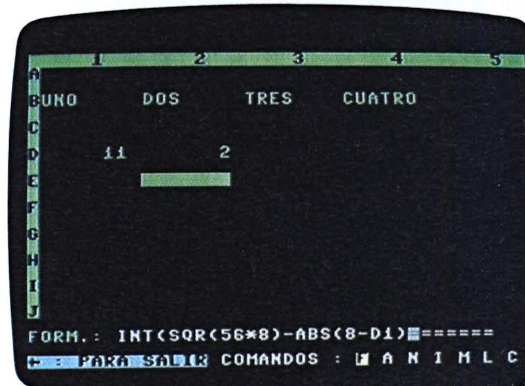


Introducción de fórmulas. Para las fórmulas se ha previsto un campo de introducción de 30 caracteres, cuyo contenido no deberá ser transferido simplemente de manera visual en el tablero, sino que deberá analizarse y procesarse para dar el resultado. Este comando puede utilizarse de diferentes maneras: introduciendo por ejemplo las coordenadas de una celda que contiene datos numéricos, éstos se toman y se insertan en la celda actual (en la foto, el contenido de D1 se ha transferido a la E2, casilla en la que se ha introducido la fórmula).

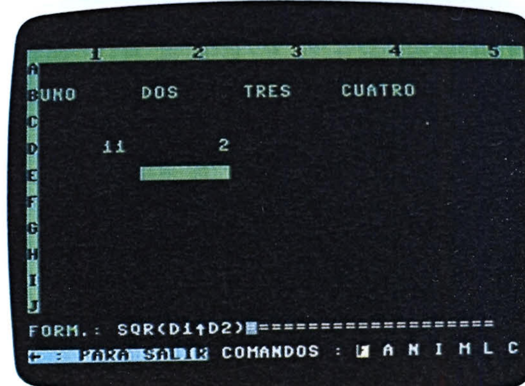


La introducción de fórmulas permite utilizar operadores matemáticos, además de varias funciones. Las introducciones reconocidas por el programa son:

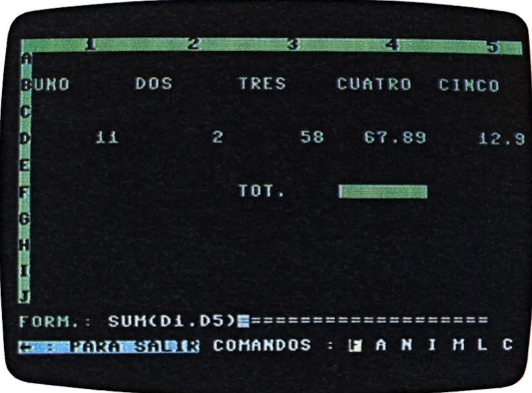
↑	= elevación a potencia	ABS	= valor absoluto
/	= división	INT	= valor entero
*	= multiplicación	COS	= coseno
–	= resta (o números negativos)	SIN	= seno
		TAN	= tangente
+	= suma	SQR	raíz cuadrada



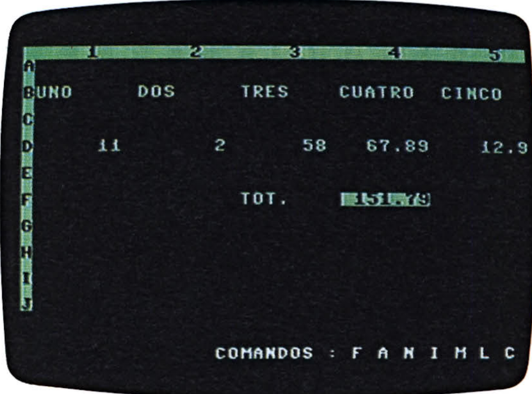
En el espacio reservado a la introducción de las fórmulas es posible insertar las coordenadas de casillas que contienen valores numéricos o ulteriores fórmulas. Este método hace que los 30 caracteres sean suficientes también para procesos complejos: la rutina de cálculo realizará la sustitución del direccionado (p.e. "D1") por los valores contenidos en aquella casilla.



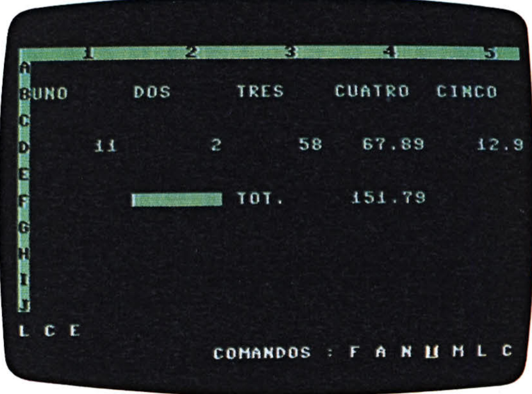
Función SUM. En la introducción de las fórmulas se ha previsto la función SUM, que permite calcular la suma de los valores contenidos en una serie de casillas del tablero. En este ejemplo se ha pedido el total de la suma de las casillas de la línea correspondiente a la letra D.



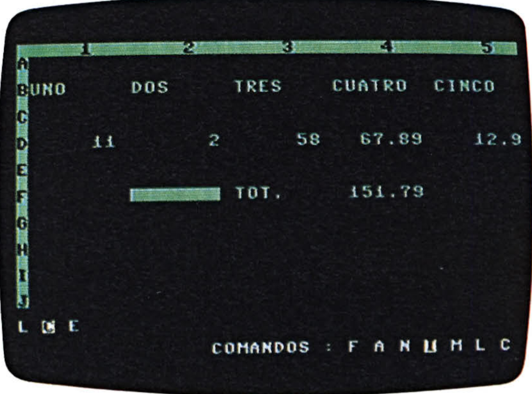
El programa, una vez reconocida la función SUM, identifica la casilla de principio para el cálculo y le suma los valores leídos en las siguientes casillas hasta la última. El resultado se presenta después en la casilla en la que se ha pedido la función (en la foto, la correspondiente a las coordenadas F4).



Comandos. Los comandos que no corresponden a la introducción de los datos son identificados por las letras I, M, L y C, y tienen los siguientes significados:
I = inserciones
M = memorización
L = carga de los datos
C = copia
En este caso se ha activado el comando I: sigue la búsqueda por parte del programa del tipo de inserción a realizar.



Se ha seleccionado el comando I (inserción) y aparecen las opciones L, C, E con los significados:
L = línea (inserta una línea)
C = columna (inserta una columna)
E = end (termina el comando).



En la máscara anterior, el usuario ha seleccionado la opción C: el sistema inserta una nueva columna en la posición indicada por el cursor.

Ejemplo de utilización del comando C (copia). La letra correspondiente aparece invertida y el programa se pone en espera de los parámetros, o sea la indicación de la zona del tablero a copiar. El primer parámetro está constituido por la línea de principio de la zona.

La foto muestra la introducción del segundo parámetro, o sea la línea en la que termina la parte a copiar. Obsérvese que la dirección a las líneas está constituida por números progresivos: 1 para la primera línea, 2 para la segunda, etc.

Ultimo parámetro del comando C: el sistema pregunta en qué línea debe empezar el posicionado de la copia.

La foto muestra el resultado del comando anterior: se han copiado las dos primeras líneas (A y B) en las posiciones G y H (correspondientes a 7 y 8).

Comando M (memorización). Este comando permite memorizar parte del tablero de un fichero. Los parámetros asociados al comando están constituidos por las direcciones (nombres de las celdas) del principio y del final de la zona a transferir. En la foto se ha presentado la celda de principio del área de memorización (celda A1).

La foto muestra la introducción de la dirección que identifica el final de la zona de memorización (celda H6).

En este punto se activa la memorización en el fichero de la zona comprendida entre la celda A1 y la celda H6. El fichero en el que se memorizan los datos es único, por lo que los datos eventualmente presentes en el mismo se pierden.

Ejemplo de aplicación para la distribución de los gastos comunitarios: en la foto se muestra la tabla de las superficies habitadas.

```
   1        2        3        4        5
A
B PISO 1     11.96 MO
C PISO 2      3.50 MO
D PISO 3     11.27 MO
E PISO 4      9.98 MO
F PISO 5     12.93 MO
G - - - - - - - - -
H TOTAL     [illegible] MO
I
J
FORM.: SUM(B2.F2)
[illegible] COMANDOS : [illegible] A N I M L C
```

Cálculo de las distribuciones milesimales: en la foto se presenta la fórmula:

(1000/H2) * C2

en la que H2 es la superficie habitada total y C2 es la superficie del apartamento considerado (por tanto, esta dirección puede variar de B2 a F2).

```
   1        2        3        4        5
A
B PISO 1     11.96 MO         240.934
C PISO 2      3.50 MO
D PISO 3     11.27 MO
E PISO 4      9.98 MO
F PISO 5     12.93 MO
G - - - - - - - - -
H TOTAL      49.64 MO
I
J
FORM.: (1000/H2)*C2
[illegible] COMANDOS : [illegible] A N I M L C
```

Tabla de la distribución milesimal completada (aplicando 5 veces la fórmula anterior): en la celda J3 se ha insertado el gasto total a repartir.

```
   1        2        3        4        5
A
B PISO 1     11.96 MO         240.934
C PISO 2      3.50 MO         70.5076
D PISO 3     11.27 MO         227.034
E PISO 4      9.98 MO         201.047
F PISO 5     12.93 MO         260.475
G - - - - - - - - -
H TOTAL      49.64 MO
I
J GASTO    TOTAL     [illegible]

                     COMANDOS : F A N I M L C
```

Cálculo de los importes debidos por cada apartamento en función del gasto anterior.

```
   1        2        3        4        5
F PISO 5     12.93 MO         260.475
G - - - - - - - - -
H TOTAL      49.64 MO
I
J GASTO    TOTAL       90000
K IMPORTE  PISO 1:  21684.0
L IMPORTE  PISO 2:  6345.68
M IMPORTE  PISO 3:  20433.0
N IMPORTE  PISO 4:  18094.2
O IMPORTE  PISO 5:  23442.7

                     COMANDOS : F A N I M L C
```

La lógica del programa

Las funcionalidades del programa pueden dividirse en dos tipos:

1 / Posicionado del cursor, introducción de datos o cálculos, con eventual ejecución, y presentación del resultado
2 / Gestión de funciones particulares como el borrado o el desplazamiento de valores de una celda a otra.

El tipo de funcionamiento pedido por el usuario se determina en el momento de la introducción de cada carácter. Normalmente, el programa se encuentra en el primer estado (desplazamiento o introducción) y si se introduce un carácter particular que se utiliza como flag (por ejemplo, el símbolo "C") pasa a la otra función. Por tanto, el programa principal (main) está constituido por un bucle de lectura, comprobación del carácter y definición de la actividad requerida. En la página siguiente se ha representado el diagrama de principio y, para analizarlo, antes hay que definir la estructura que se va a utilizar.
Como se ha indicado, el espacio disponible está dividido en celdas, cada una identificada por una línea y una columna propias que deben ser presentadas al usuario. Por tanto, la primera función a activar es la presentación de un sistema de referencia constituido por dos ejes con origen en la parte superior izquierda y una serie de símbolos de identificación de la posición. En esta versión, la línea está identificada por una letra, mientras que la columna lo está por un número (hay que tener en cuenta que por el término "columna" no se entiende una columna de la pantalla, sino una columna de celdas, o sea un grupo de columnas de pantalla en número igual a la longitud en bytes de cada celda).
Como se ha indicado, la primera función es la presentación de los ejes (subrutina 8000) que, además, pone L = 1 y C = 1, indentificando así la primera celda (en el cruce de la línea 1 con la columna 1). En este punto, el programa se pone en espera de un carácter y, en función del tipo de carácter introducido, se tiene la activación del correspondiente bloque principal:

1 / Subrutina para los desplazamientos
2 / Subrutina para los comandos
3 / Subrutina para la introducción.

Al final de cada función, el programa vuelve a la instrucción de espera de un carácter.

Subrutina de cálculo. Esta subrutina sólo se examina desde el punto de vista funcional; para el detalle de las operaciones simples hay que consultar los listados. Para su funcionamiento hay que definir:

OP$(5) = símbolos de los cálculos aritméticos en orden de prioridad (↑, /, ∗, +, –)
FZ$(∗) = símbolos (primeras tres letras del nombre) de las funciones previstas.

En la llamada, la cadena A$ contiene el cálculo a realizar con todos los valores ya preparados, por lo que antes de activarla en la fórmula, se hace referencia a una celda de la que hay que tomar el valor y sustituirlo por la dirección. En la cadena pueden emplearse paréntesis, funciones propias del Basic y funciones particulares específicas de esta aplicación, porque las direcciones ya están decodificadas, o sea sustituidas por el correspondiente valor numérico. Por ejemplo, si se quiere sumar el contenido de la celda n.° 1 al de la n.° 3, respectivamente de nombres A1 y C1, el usuario escribirá la fórmula A1 + C1. El sistema debe interpretar las letras A1 y C1 como direcciones, tomar su contenido y transformar la cadena sustituyendo los valores numéricos por las referencias. Este proceso inicial es necesario para obtener una expresión que contenga letras que sólo indican nombres de función. Como ya se ha indicado, la subrutina puede realizar cálculos que contienen funciones. Las previstas en las distintas versiones se indican en las correspondientes instrucciones DATA de carga en la matriz FS$(∗). Téngase en cuenta que, en algunos casos, se han previsto funciones que no están incluidas en el Basic, pero muy útiles en este tipo de aplicación. En la página 2421 se ha representado el diagrama de principio de la subrutina. Los bloques principales son dos: el primero dedicado al desarrollo de las funciones y el segundo a los cálculos aritméticos. Por tanto, para eventuales implantaciones, hay que modificar el correspondiente bloque. En la salida, el resultado de la expresión se transforma en A$, que por lo tanto se utiliza tanto como parámetro de entrada como de salida.

Desplazamiento del cursor. Para analizar el funcionamiento de esta rutina hay que seguir el funcionamiento del programa desde el principio.
Como ya se ha indicado, la primera función realizada es la presentación de los ejes y de las direcciones, líneas y columnas de las celdas. En esta fase, el cursor está posicionado en la línea 1, columna 1.
Introduciendo el comando de desplazamiento, por ejemplo a la derecha, el cursor debe posicionarse en la segunda celda de la primera línea, o sea en la columna 8 de la pantalla (una celda ocupa 7 bytes). Por tanto, el posicionado del cursor en el sentido horizontal se hace con paso 7. En los diagramas se entiende con el término "columna" una columna del tablero, o sea una celda, igual a 7 columnas de vídeo.
La subrutina empieza con la lectura de la celda de

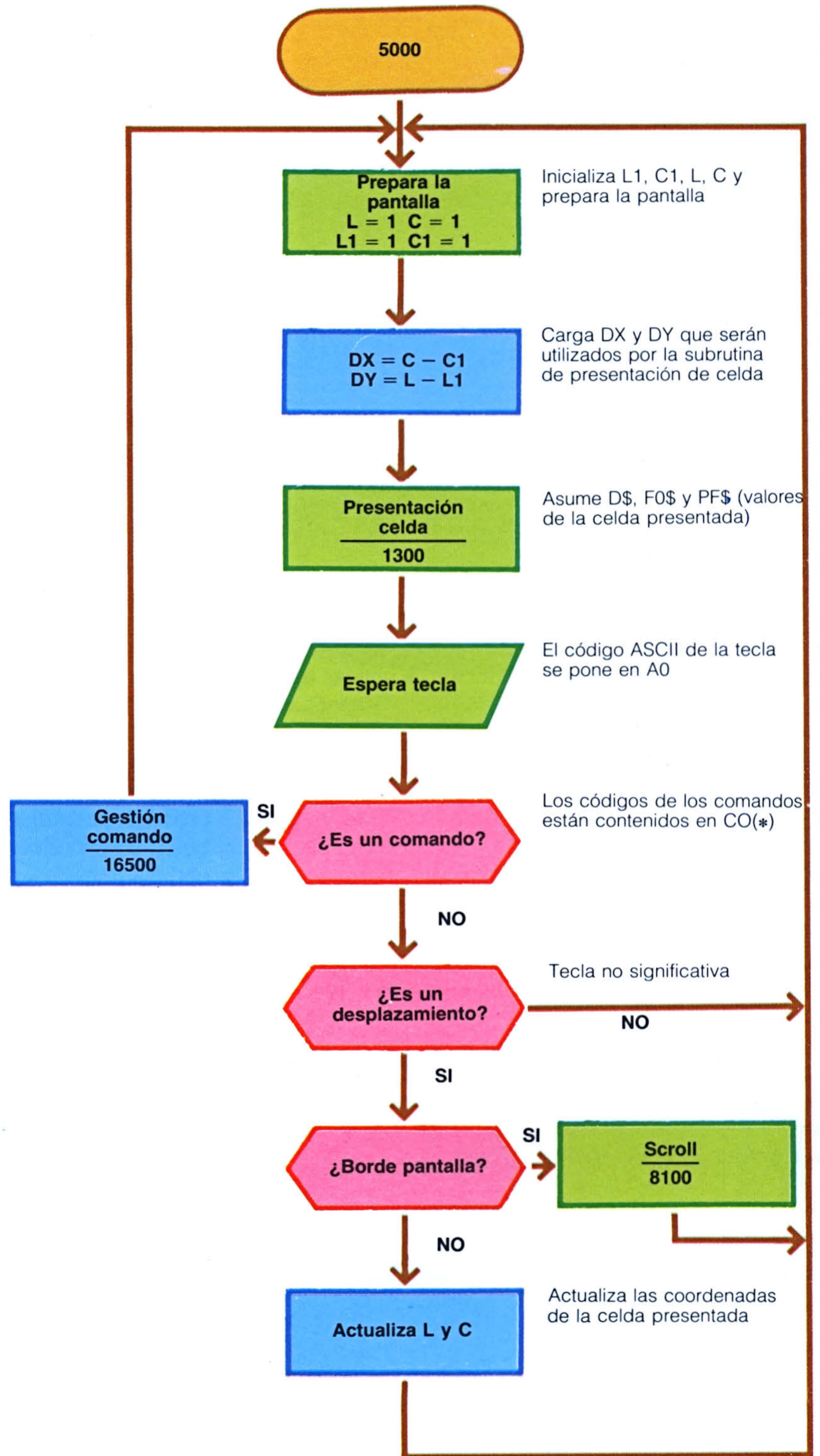
coordenadas L,C correspondientes a la posición del cursor. En la salida de la rutina de lectura se tiene:

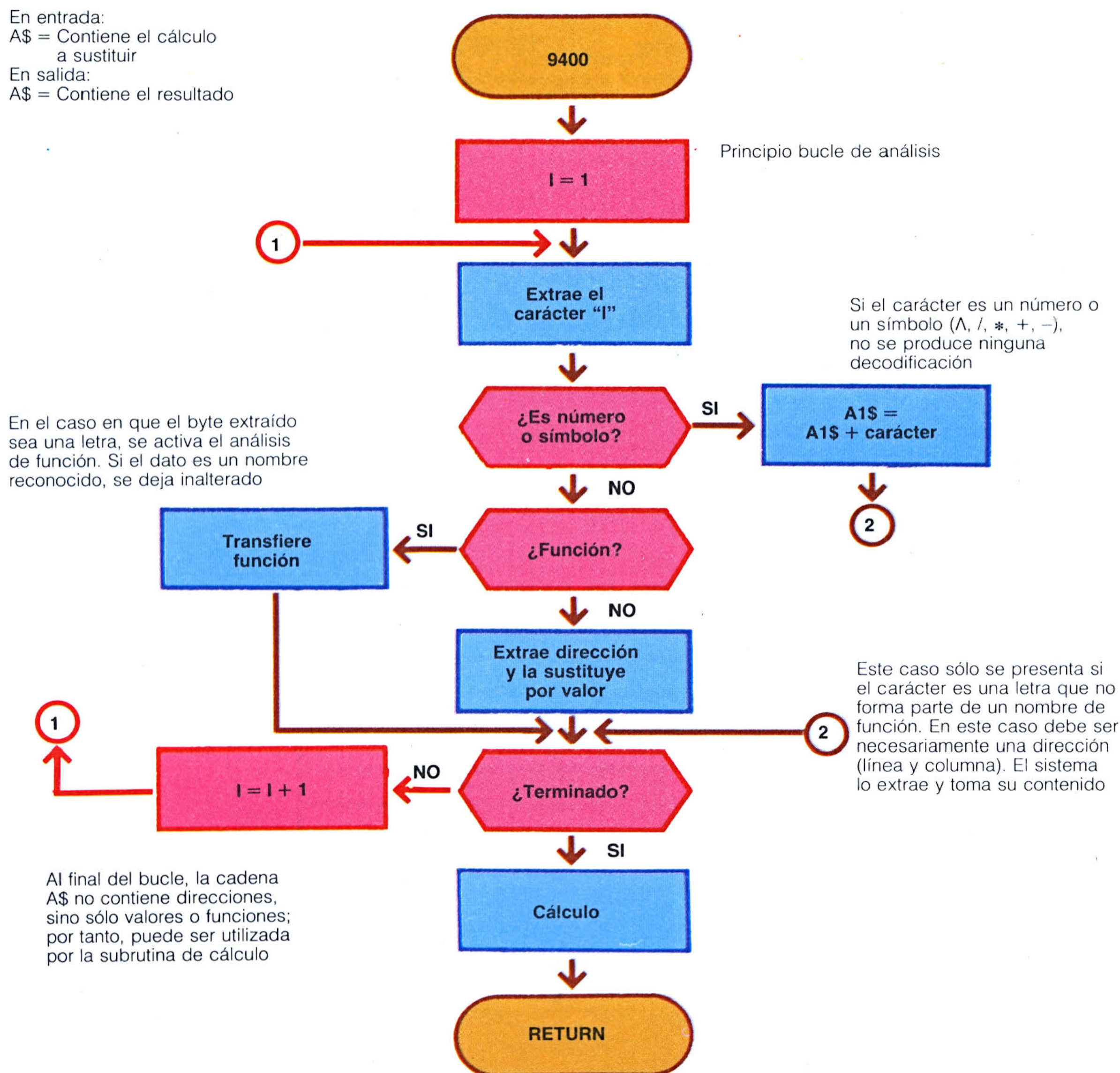
F0$ = Indicador del tipo de campo (1 byte)
PF$ = Puntero a la fórmula (1 byte)
D$ = Dato (7 bytes)

Por tanto, puede determinarse si la celda contiene o no un cálculo.
En el primer caso (PF$ < > "0"), hay que llamar la subrutina 17000 que efectúa el cálculo y deposita el resultado en la celda. En el segundo, basta con presentar el contenido.
En uno y otro caso continúa el cálculo de las nuevas coordenadas (en base al tipo de desplazamiento pedido).

Lectura del contenido de una celda. La subrutina debe restituir el contenido de la celda apuntada por los valores L y C, en las variables F0$, PF$ y D$. Por tanto, la primera función es el cálculo del byte de principio de la celda número "C" en la cadena "L" (B$(R)). Siguen la extracción de todos los datos de la celda y su desempaquetado en las tres variables de cadena.

Desarrollo de los cálculos. Los cálculos se desarrollan utilizando la subrutina generalizada descrita al principio. Sin embargo, antes de llamarla, hay que preparar los datos, o sea la fórmula, en la variable A$ y, en la salida, transferir el resultado a D$ para la escritura en la celda y la presentación en pantalla.
La subrutina no presenta ninguna dificultad ni particularidad, puesto que sólo pide el control sobre la longitud de D$. Esta función no se indica en el diagrama, y se reduce a añadir un número de espacios igual a los bytes que faltan en el caso de una longitud inferior a





la prevista, o a truncar el resultado en el caso de una longitud superior. Recuérdese que en el caso de que se añadan espacios para completar el campo, la suma debe realizarse a la izquierda del valor numérico.

Escritura de una celda. La transferencia de un valor (campo D$ + FØ$ + PF$) a una celda se obtiene insertándolo en la correspondiente posición del buffer de línea.

El tablero está compuesto por un cierto número de buffers (B$(*)), cada uno correspondiente a una línea y dividido lógicamente en tantos campos como cuantas columnas hay. Por tanto, para cambiar el valor de una celda es necesario insertar, en el correspondiente buffer de la línea a la que pertenece la celda, el valor dado en la posición correspondiente a la columna. Esta función se obtiene aislando la parte de B$(*) a la izquierda de la celda, o mejor, de los bytes que debe-

rán acoger el contenido, y a la de la derecha. Así se obtienen 3 cadenas:

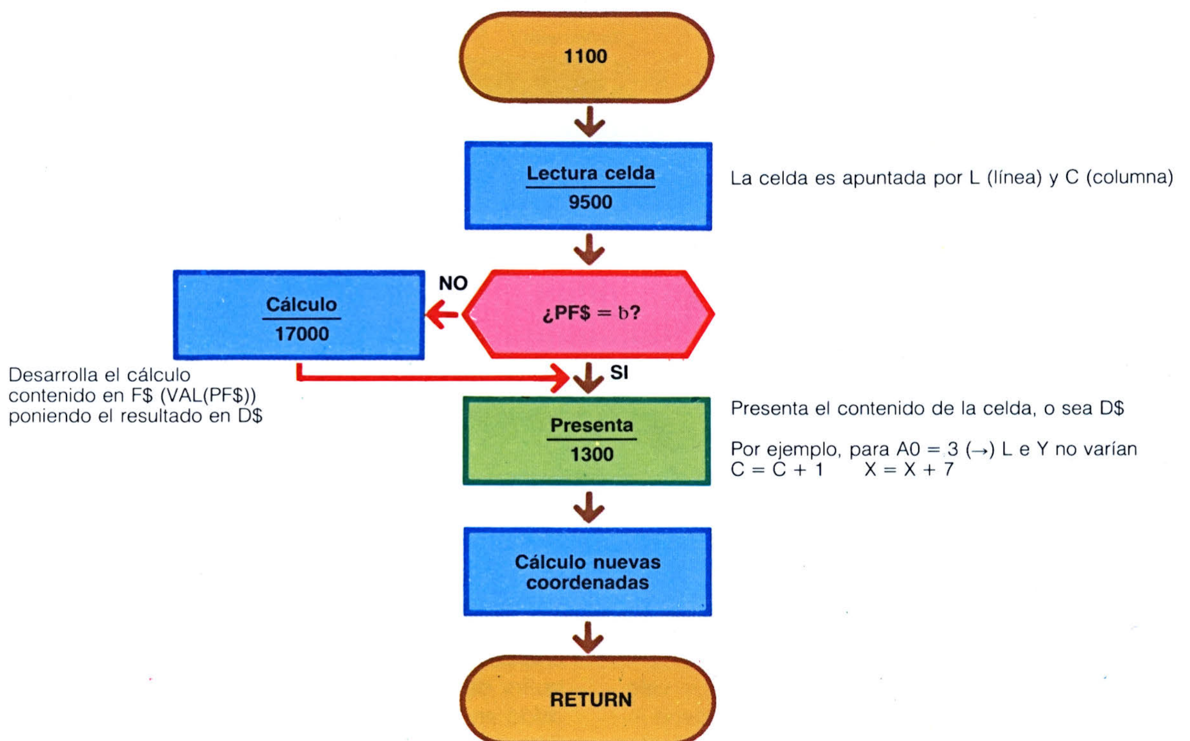
A1$= Parte de B$(*) a la izquierda del campo
A2$= Parte de B$(*) a la derecha del campo
A$ = Campo (contenido de la celda) a insertar
(FØ$ + PF$ + D$)

Sumando los tres valores se reconstruye el buffer B$(R), incluidos los nuevos valores correspondientes a la celda que hay en la columna C.
Esta subrutina, cuya lógica no queda alterada, es utilizada por todos los puntos del programa que piden la memorización del contenido de una celda (por ejemplo, en la fase de introducción o al final del desarrollo de un cálculo).
Obsérvese que al transferir el valor D$ también se reescriben los flags (FØ$, PF$ y eventualmente otros), por lo que la cadena A$ que representa el nuevo contenido de la celda debe estar compuesta por FØ$ + + PF$ + D$, cada uno con el adecuado número de bytes (1, 1, 7) para no alterar la longitud total de la celda.

Presentación del contenido de una celda. La subrutina presupone que el cursor está correctamente posicionado, o sea en la línea L y la columna C del tablero, en la correspondiente columna de pantalla: 1 + 9 * (C-1).
En la llamada sólo es necesario indicar la dirección (L, C) de la celda.
La subrutina ejecuta las funciones:

1 / Lectura del contenido de la celda (D$)
2 / Presentación del valor.

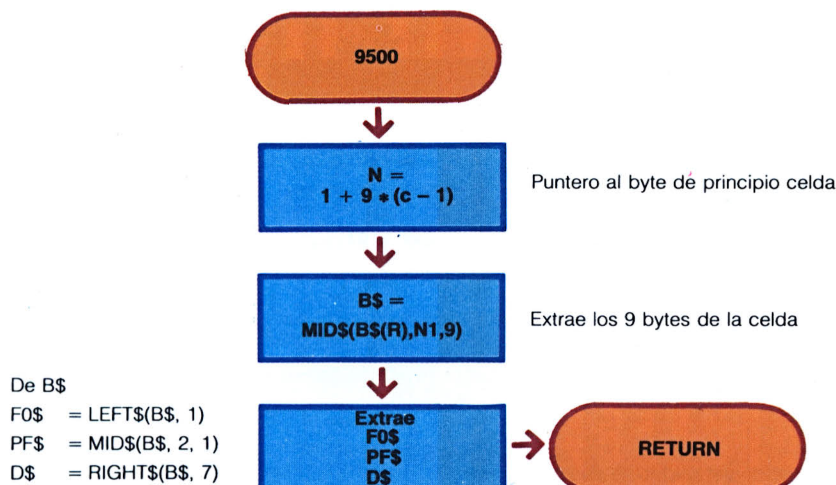
En el diagrama se muestran los dos modos posibles de determinación del tipo de desplazamiento.
El primero se basa en el contenido del flag H1, implantado en otro lugar, para vincular el desplazamiento independientemente del tipo de dato.
El segundo analiza el contenido de la celda (flag FØ$) y activa el desplazamiento congruente (izquierda para los datos alfanuméricos, derecha para los numéricos).
El flag H1, cuya activación genera el desplazamiento independientemente del tipo de dato, se implanta a



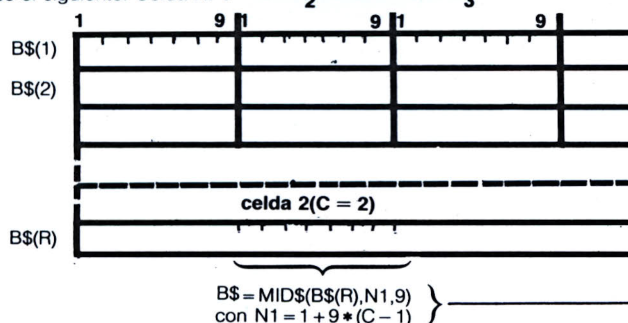
LECTURA DEL CONTENIDO DE UNA CELDA

En llamada:
L, C = Coordenadas de la celda

En respuesta:
F0$ = Tipo del campo
PF$ = Puntero a la fórmula
D$ = Contenido de la celda



Se recuerda que el formato es el siguiente: Celda N. 1 2 3



través de la gestión de los comandos (subrutina 16000) y se utiliza para presentaciones especiales.

En el diagrama, los bloques finales, que generan el desplazamiento deseado, no están detallados porque dependen del tipo de máquina utilizado. Sólo hay que tener en cuenta que la variable local H2 se ha insertado para no modificar el flag H1, cuyo contenido es gestionado por otra vía.

Traslación de la pantalla vídeo. En la página 2439 se ha representado el diagrama de principio que muestra la lógica seguida para asignar los valores de los extremos del bucle y el método seguido.

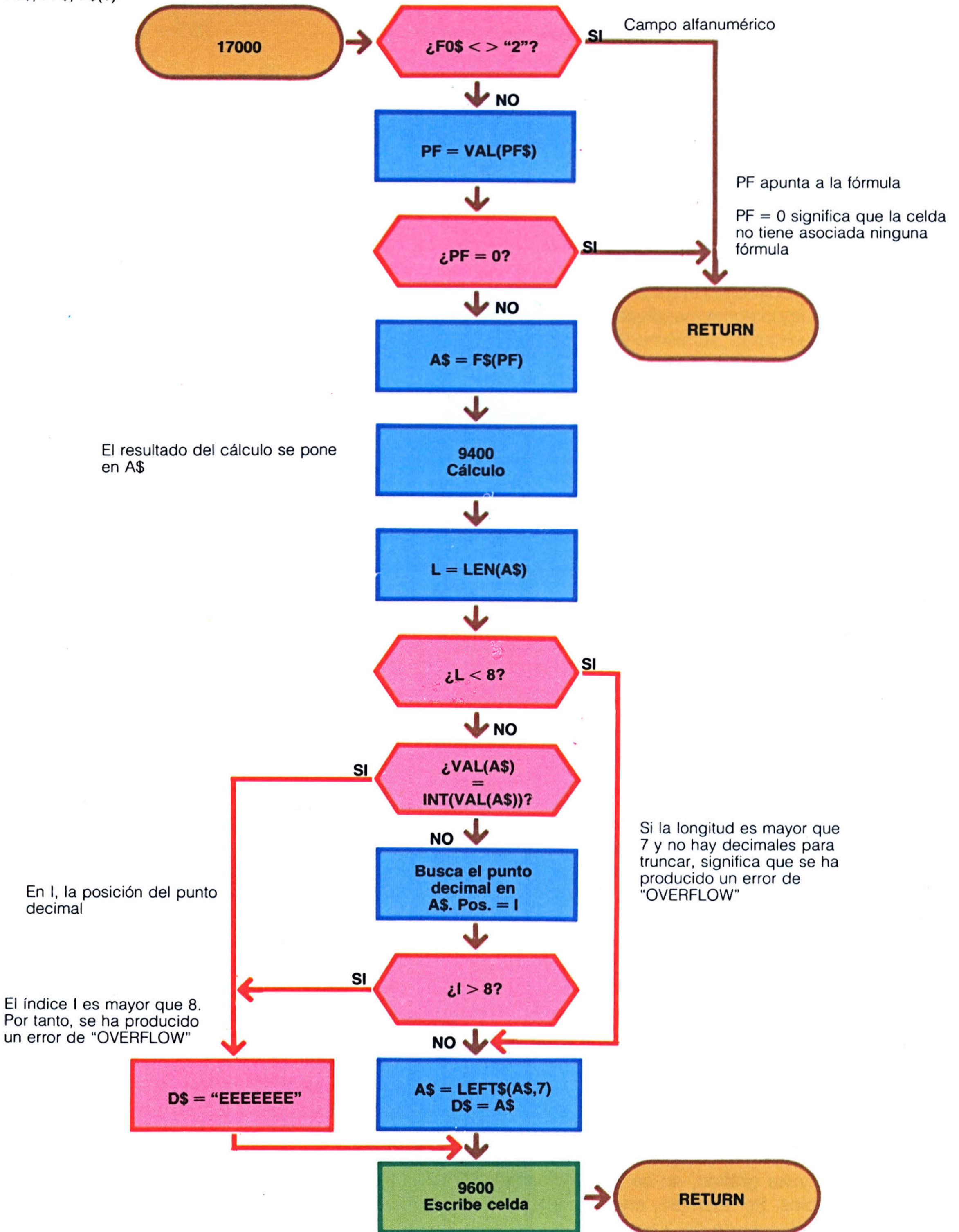
El diagrama está muy simplificado y no corresponde a los listados porque, para no complicarlo demasiado desde el punto de vista gráfico, no se han indicado algunos de sus bucles. Por ejemplo, en el desplazamiento vertical (parte izquierda del flujo), hay indicado el bucle sobre las líneas, pero la presentación de las celdas sólo se indica como subrutina. En realidad se trata de un bucle similar al mostrado en el flujo de la derecha (desplazamiento horizontal).

Análogamente, en el flujo de la derecha se muestra el bucle que permite la traslación horizontal de una línea, mientras que en los listados se necesita uno más exterior sobre todas las líneas presentadas.

Finalmente, hay que observar que la metodología seguida, consistente en simular un «scroll» del vídeo en lenguaje Basic, es la menos adecuada desde el punto de vista funcional, aunque sí la más comprensible.

Seguramente, una mejor versión puede obtenerse sustituyendo esta rutina (8100) por una en lenguaje máquina que trabaje sobre la memoria vídeo, transfiriéndole cada vez la zona de B$(*) a presentar.

Como esta solución está estrechamente ligada a la máquina utilizada, no se ha representado en el diagrama, y en su lugar se ha indicado la de la página 2439, que únicamente puede ser útil a nivel funcional.



DESARROLLO CALCULOS
Entradas:
L,C = Coordenadas celda
F0$, PF$, F$(*)
17000
¿F0$ < > "2"?
SI
Campo alfanumérico
NO
PF = VAL(PF$)
PF apunta a la fórmula
PF = 0 significa que la celda no tiene asociada ninguna fórmula
¿PF = 0?
SI
NO
RETURN
A$ = F$(PF)
El resultado del cálculo se pone en A$
9400
Cálculo
L = LEN(A$)
¿L < 8?
SI
NO
SI
¿VAL(A$) = INT(VAL(A$))?
NO
Si la longitud es mayor que 7 y no hay decimales para truncar, significa que se ha producido un error de "OVERFLOW"
En I, la posición del punto decimal
Busca el punto decimal en A$. Pos. = I
SI
¿I > 8?
El índice I es mayor que 8. Por tanto, se ha producido un error de "OVERFLOW"
NO
D$ = "EEEEEEE"
A$ = LEFT$(A$,7)
D$ = A$
9600
Escribe celda
RETURN



El tablero electrónico/Versión Commodore 64

```
1 REM *********************
2 REM *                   *
3 REM * TABLERO VERS C-64 *
4 REM *                   *
5 REM *********************
8 PRINT CHR$(142)
10 DIM OP$(5),FZ$(6),CO(9),B$(15)
15 RESTORE
20 FOR I=1 TO 5:READ OP$(I):NEXT I
25 FOR I=1 TO 6:READ FZ$(I):NEXT I
30 FOR I=1 TO 7:READ CO(I):NEXT I
35 FOR I=1 TO 7:READ TT(I):NEXT I
500 REM PREPARACIONES
505 HO$="J":PRINTHO$:POKE53280,0:POKE53281,0
507 POKE198,0
510 POKE646,5:POKE649,15:POKE650,128:POKE657,128
512 B$="          ":BA$="":BU$="":ZE$=""
515 FOR I=1 TO 4:BA$=BA$+B$:BU$=BU$+B$:NEXT I
517 B$=""
520 FORI=1TO15:ZE$=ZE$+"0000":FORJ=1TO24:B$(I)=B$(I)+LEFT$(BA$,10):NEXTJ,I
525 RV=1:NR=0
530 F1$="========================================"
550 FC=1:LD=0:GOTO 5000
1200 REM *** ESPERA TECLA ***
1202 POKE 198,0
1205 GET A$:IF A$="" THEN 1205
1210 A0=ASC(A$)
1215 KK=1
1220 KK=KK+1:IF KK=8 THEN 1250
1225 IF A0=TT(KK) THEN 1202
1230 GOTO 1220
1250 RETURN
1300 REM ENCIENDE CELDA
1310 POKE 199,NR:X=XX:Y=YY:GOSUB11000:PRINT DD$
1320 XX=DX*8+2:YY=DY*2+2
1330 GOSUB 9500:POKE 199,RV:IF PF$<>"0" AND PF$<>" "THEN GOSUB17000
1340 X=XX:Y=YY:GOSUB11000:PRINT D$:DD$=D$
1350 POKE 199,NR:RETURN
2900 REM ***  ELIMINA ESPACIOS  ***
2910 Z2$="":A=LEN(Z$):FORI=1TOA:Z1$=MID$(Z$,I,1):IFZ1$<>" "THENZ2$=Z2$+Z1$
2920 NEXT I:Z$=Z2$:RETURN
3000 REM BORRA LINEA YT
3010 X=1:Y=YT:GOSUB11000:PRINTLEFT$(BA$,39)
3020 RETURN
3500 REM ***** ENTRADA CAMPO *****
3502 X1=X+1:X2=X+L0-1:POKE 650,0:POKE 646,15
3505 FI$=C$+MID$(F1$,1,L0-LEN(C$))
3510 GOSUB12000:PRINTFI$:GOSUB12000
3515 POKE 198,0:POKE 204,0:POKE 199,0
3520 IF PEEK(646)<>15 THEN POKE646,15
3521 GET H$:IF H$="" THEN 3521
3522 IF ASC(H$)=44 THEN 3520
3525 A=ASC(H$):IF A=13 THEN 3600
3526 IF A=95 THEN 3660
3527 IF A=29 OR A=157 THEN 3535
3530 IF A<32 OR A>94 THEN 3520
3535 LC=1024+(40*Y)+X:L1=PEEK(LC):IF L1=160 THEN L1=32
3540 IF L1>128 THEN L1=L1-128
3545 IF A<>29 AND A<>157 THEN 3575
3550 IF A=157 THEN 3565
3555 IF X>=X2 THEN 3520
3560 POKE LC,L1:X=X+1:GOSUB12000:GOTO 3520
3565 IF X<X1 THEN 3520
3570 POKE LC,L1:X=X-1:GOSUB12000:GOTO 3520
3575 REM
3585 GOSUB12000:PRINTH$;:X=X+1
3590 IF X=>X2 THEN X=X2:GOSUB12000:GOTO 3520
3595 GOTO 3520
3600 REM
3605 C$=""
3610 FOR K=X1-1 TO K+L0-1:K1=1024+(40*Y)+K
3615 AQ=PEEK(K1)
3620 IF AQ=160 THEN AQ=32
3622 IF AQ>128 THEN AQ=AQ-128
3625 IF AQ=>1 AND AQ<=30 THEN AQ=AQ+64
3630 C$=C$+CHR$(AQ)
3635 NEXT K
3640 FOR J=1 TO L0
3645 G$=MID$(C$,J,1):IF G$="=" THEN C$=LEFT$(C$,J-1)+" "+RIGHT$(C$,L0-J)
3650 NEXT J
3660 POKE 204,1:POKE 650,128:POKE 646,5
3680 RETURN
5000 REM *************
5001 REM *    MAIN   *
5002 REM *************
5010 GOSUB8000:DD$="":XX=1:YY=1
5020 DX=C-C1:DY=R-R1
5030 GOSUB 1300:GOSUB 1200
5040 IF A0<>29 THEN 5070
5050 IF DX=4 AND C1<22 THEN C=C+1:C1=C1+1:GOSUB 8100:GOTO 5020
5060 IF DX<4 THEN C=C+1:GOTO 5020
5070 IF A0<>157 THEN 5100
5080 IF DX=0 AND C1>1 THEN C=C-1:C1=C1-1:GOSUB 8100:GOTO 5020
5090 IF DX>0 THEN C=C-1:GOTO 5020
5100 IF A0<>17 THEN 5130
5110 IF DY=9 AND R1<6 THEN R=R+1:R1=R1+1:GOSUB 8100:GOTO 5020
5120 IF DY<9 THEN R=R+1:GOTO 5020
5130 IF A0<>145 THEN 5160
5140 IF DY=0 AND R1>1 THEN R=R-1:R1=R1-1:GOSUB 8100:GOTO 5020
5150 IF DY>0 THEN R=R-1:GOTO 5020
5160 GOTO 16000
5165 GOTO 5020
6000 REM *** GESTION DISCO ***
6002 OPEN1,8,15:RETURN
6004 REM
6006 INPUT#1,AD,X$,CD,DD
6008 IF AD=72ORAD=74ORAD=70ORAD=26 THENCLOSE1:CLOSE2:GOTO 7000
6009 IFAD=62ANDLD=1THENB$="EL FICHERO DATOS NO ESTA":CLOSE1:CLOSE2:GOTO7025
6010 RETURN
6500 REM **** CREACION FICHERO DATOS ****
6502 AA$=""
6504 FOR J9=1 TO 234:AA$=AA$+"#":NEXT J9
6506 OPEN1,8,15
6508 OPEN2,8,3,"0:TTBB,L,"+CHR$(235)
6510 FOR J9=1 TO 24
6515 PA=J9:GOSUB 6800:PRINT#2,AA$
6520 NEXT J9
6525 INPUT#1,AD,X$,CD,DD
6530 CLOSE1:CLOSE2
6535 RETURN
6540 REM ----------- ↑ --------------------------
6800 REM * APUNTA AL REGISTRO *
6805 R1=PA:R2=0
6810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6815 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6820 RETURN
7000 REM * ERROR DE UNIDAD DE DISCO *
7003 IF AD=26 THEN B$="ESCRITURA IMPOSIBLE":GOTO 7025
7005 IF AD=72 THEN B$="ESPACIO EN DISCO AGOTADO":GOTO 7025
7007 IF AD=74 THEN B$="EL DISCO NO ESTA":GOTO 7025
7009 IF AD=70 THEN B$="CANAL NO DISPONIBLE":GOTO 7025
7012 B$="FICHERO DATOS NO UTILIZABLE"
7025 Y=22:X=1:GOSUB11000:PRINT"■";B$;"■";
7030 POKE198,0:WAIT198,1:YT=22:GOSUB3000
7035 ED=1:RETURN
8000 REM **** INICIALIZA PANTALLA ****
8010 POKE646,5:R$="":FOR I=65 TO 79:R$=R$+CHR$(I):NEXT I:POKE 199,RV
8015 V$="FANIMLC"
8020 R1=1:C1=1:R=1:C=1:FOR I=2TO20:X=1:Y=I:GOSUB11000:PRINT" ";:NEXT I
8022 X=1:Y=24:GOSUB11000:PRINT"■                    ";
8025 POKE 199,NR:X=16:Y=24:GOSUB11000:PRINT"■COMANDOS : F A N I M L C■";
8100 REM ***** SCROLL *****
```

```
8110 POKE 199,RV:C2=C1:X=1:Y=1:GOSUB11000:PRINTLEFT$(BA$,40);
8120 FOR I=5 TO 37 STEP 8:POKE 199,RV
8130 X=I:GOSUB11000:PRINTC2:C2=C2+1:NEXT I
8140 R2=R1:FOR I=2 TO 20 STEP 2
8150 X=1:Y=I:GOSUB11000:PRINT"■"MID$(R$,R2,1):R2=R2+1
8160 NEXT I:POKE 199,NR
8170 Y=2:FOR I=R1 TO R1+9
8180 X=2:FOR J=C1 TO C1+4
8190 N1=1+9*(J-1):A$=MID$(B$(I),N1,9)
8195 FO$=LEFT$(A$,1):PF$=MID$(A$,2,1)
8200 D$=RIGHT$(A$,7):GOSUB11000
8210 PRINTD$:X=X+8:NEXT J
8220 Y=Y+2:NEXT I:RETURN
9000 REM *** CAL 2 ***
9005 ER=0:L=LEN(CA$):IF L<3 THEN 9030
9010 FORI=2TOL-1:CH$=MID$(CA$,I,1):FORJ=1TO5:IF OP$(J)=CH$ THEN 9020
9015 NEXT J,I:ER=1:GOTO 9030
9020 CH$=MID$(CA$,I+1,1):IF CH$="*" OR CH$="↑" OR CH$="/" THEN ER=2:GOTO9030
9022 N1=VAL(MID$(CA$,1,I-1)):N2=VAL(MID$(CA$,I+1,40))
9023 ON J GOTO 9035,9040,9050,9055,9060
9024 CA$=STR$(N1):L9=LEN(CA$)-1
9027 IF N1<0 THEN 9030
9028 CA$=RIGHT$(CA$,L9):IF N1>0 THEN CA$="+"+CA$
9030 GOTO 9120
9035 N1=N1↑N2:GOTO9024
9040 IF N2=0 THEN ER=3:GOTO9024
9045 N1=N1/N2:GOTO9024
9050 N1=N1*N2:GOTO9024
9055 N1=N1-N2:GOTO9024
9060 N1=N1+N2:GOTO9024
9065 REM *** **** ***
9070 ER=0:LO=LEN(FO$):IF LO<3 THEN RETURN
9075 FOR OP=1 TO 5:PO=0:P1=0:P2=0:FOR IO=2 TO LO-1:CH$=MID$(FO$,IO,1)
9077 CH=ASC(CH$):IF CH$=OP$(OP) AND PO=0 THEN PO=IO:GOTO 9095
9080 O=0:IF (CH=94ORCH<48)AND CH<>46 THEN O=1
9085 IF O=1 AND PO=0 THEN P1=IO
9090 IF O=1 AND PO>0 AND IO-PO>1 THEN P2=IO:GOTO 9105
9095 NEXT IO:IF PO>0 THEN 9105
9100 NEXT OP:RETURN
9105 IF P1=0 THEN P1=1:GOTO 9115
9110 CH$=MID$(FO$,P1,1):IF CH$<>"+" AND CH$<>"-"THEN P1=P1+1
9115 P2=P2-1:IF P2<0 THEN P2=LO
9117 CA$=MID$(FO$,P1,P2-P1+1):GOTO 9000
9120 IF ER<>0 THEN RETURN
9125 GOSUB 9280:FO$=MID$(FO$,1,P1-1)+CA$+MID$(FO$,P2+1,40):GOTO 9070
9130 REM *** **** ***
9140 ER=0:LA=LEN(A$):IF LA<3 THEN 9165
9145 PI=0:PF=0:FOR IA=1 TO LA:CH$=MID$(A$,IA,1):IFCH$="("THENPI=IA
9150 IF CH$=")" THEN 9185
9155 NEXT IA:IF PI>0 THEN ER=4:RETURN
9160 FO$=A$:GOSUB 9065:A$=FO$
9165 IF VAL(A$)=0 THEN A$="0":RETURN
9175 IF LEFT$(A$,1)="+" THEN A$=MID$(A$,2,80)
9180 RETURN
9185 IF PI=0 THEN ER=4:RETURN
9190 PF=IA:FO$=MID$(A$,PI+1,PF-PI-1):GOSUB 9065:IF ER<>0 THEN RETURN
9195 IF PI>3 THEN GOSUB 9220
9200 IF PI<=1 THEN 9210
9205 IF A<0ANDMID$(A$,PI-1,1)="-"THENPI=PI-1:FO$="+"+MID$(FO$,2,80)
9210 IF ER<>0 THEN RETURN
9215 A$=MID$(A$,1,PI-1)+FO$+MID$(A$,PF+1,80):GOTO 9140
9220 REM *** **** ***
9225 ER=0:CH$=MID$(A$,PI-1,1):IF CH$<"A"ORCH$>"Z"THENRETURN
9230 FU$=MID$(A$,PI-3,3):I=0
9231 I=I+1:IF I>6 THEN ER=5:RETURN
9234 IF FU$=FZ$(I)THEN 9240
9235 GOTO 9231
9240 A1=VAL(FO$):ON I GOTO 9245,9250,9255,9260,9265,9275
9241 FO$=STR$(A1):CA$=FO$:GOSUB9280:FO$=CA$
9242 L9=LEN(FO$)-1:IF A1<0 THEN 9244
9243 FO$=RIGHT$(FO$,L9)
9244 PI=PI-3:RETURN
9245 A1=ABS(A1):GOTO 9241
9250 A1=INT(A1):GOTO 9241
9255 A1=COS(A1):GOTO 9241
9260 A1=SIN(A1):GOTO 9241
9265 IF A1<0 THEN ER=6:GOTO 9241
9270 A1=SQR(A1):GOTO 9241
9275 A1=TAN(A1):GOTO 9241
9280 REM *** **** ***
9290 L=LEN(CA$):IC=0
9291 IC=IC+1:IF IC>L THEN RETURN
9292 IF MID$(CA$,IC,1)="E" THEN 9300
9295 GOTO 9291
9300 N$=MID$(CA$,2,IC-2):L=LEN(N$)
9301 I=0
9303 I=I+1:IF I>L THEN 9310
9305 IF MID$(N$,I,1)="."THENP=I:GOTO9315
9307 GOTO 9303
9310 P=L
9315 E=VAL(MID$(CA$,IC+1,3)):IF E<0 THEN 9355
9320 DD=L-P:IFDD=0THEN9345
9325 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80)
9330 IF DD<=E THEN 9345
9335 N$=MID$(N$,1,P+E-1)+"."+MID$(N$,P+E,80):DD=E
9345 N$=N$+MID$(ZE$,1,E-DD):GOTO 9375
9355 IF P=L THEN P=P+2-L:GOTO 9365
9360 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80):L=1
9365 N$=MID$(ZE$,1,-E-L+1)+N$
9370 N$=MID$(N$,1,P-1)+"."+MID$(N$,P,80)
9375 CA$=LEFT$(CA$,1)+N$:RETURN
9400 REM *** **** ***
9405 Z$=A$:GOSUB 2900:A$=Z$
9410 LA=LEN(A$):IF LEFT$(A$,3)="SUM" THEN GOTO 14000
9415 FOR I=1 TO LA
9420 CH$=MID$(A$,I,1):IF CH$>="A"ANDCH$<="Z"THEN 9440
9430 NEXT I:GOTO 9130
9440 PI=I:C$=MID$(A$,I+2,1):IF C$>="A"ANDC$<="Z"THENI=I+2:GOTO9430
9450 IFCH$>"O" THEN ER=7:RETURN
9460 R=ASC(CH$)-64:C=VAL(MID$(A$,PI+1,1)):IF C=0 THEN ER=7:RETURN
9470 GOSUB 9500:Z$=D$:GOSUB 2900
9480 A$=MID$(A$,1,PI-1)+Z$+MID$(A$,PI+2,80)
9490 GOTO 9410
9500 REM LECTURA CELDA
9510 N1=1+9*(C-1):B$=MID$(B$(R),N1,9)
9520 F0$=LEFT$(B$,1):PF$=MID$(B$,2,1)
9530 D$=RIGHT$(B$,7):RETURN
9600 REM ESCRITURA CELDA
9610 N1=9*(C-1):N2=N1+10
9620 A1$=MID$(B$(R),1,N1):A2$=MID$(B$(R),N2,235)
9630 D$=LEFT$(D$,7)
9640 IF F0$="1" THEN D$=D$+MID$(BA$,1,7-LEN(D$))
9650 IF F0$="2" THEN D$=MID$(BA$,1,7-LEN(D$))+D$
9660 B$(R)=A1$+F0$+PF$+D$+A2$:RETURN
10000 REM DATOS
10005 DATA "↑","/","*","-","+"
10010 DATA "ABS","INT","COS","SIN","SQR","TAN"
10015 DATA 70,65,78,73,77,76,67
10020 DATA 18,19,20,147,148,14,142
11000 REM POSICIONADO CURSOR
11002 IF X<=0 OR Y<=0 THEN STOP
11005 POKE211,X-1:POKE214,Y-1
11010 SYS58732
11015 RETURN
12000 REM POSICIONADO CURSOR II
12002 IF X<0 OR Y<0 THEN STOP
12005 POKE211,X:POKE214,Y
12010 SYS58732
12015 RETURN
14000 REM FUNCION 'SUM'
14005 ER=0
14010 FOR I=5 TO LA-1
14020 IF MID$(A$,I,1)="," THEN 14040
14030 NEXT I:ER=10:A$="ERR=10":RETURN
14040 A1$=MID$(A$,5,I-5):A2$=MID$(A$,I+1,LA-I-1)
14060 RI$=LEFT$(A1$,1):RF$=LEFT$(A2$,1)
14080 CI=VAL(MID$(A1$,2,2)):CF=VAL(MID$(A2$,2,2))
14090 RI=0:RF=0:A1=0
14100 FOR I=1 TO 15:CH$=MID$(R$,I,1)
14110 IF CH$=RI$ THEN RI=I
14120 IF CH$=RF$ THEN RF=I
14130 NEXT I
14140 IF RI=0 OR RF=0 OR CI=0 OR CF=0 THEN ER=10:A$="ERR=10":RETURN
14145 IF RI<>RF AND CI<>CF THEN RI=0:GOTO 14140
14150 IF CI=CF THEN 14190
14160 R=RI:FOR I=CI TO CF:C=I
14170 GOSUB 9500:A1=A1+VAL(D$)
14180 NEXT I:GOTO 14220
```

```
14190 C=CI:FOR I=RI TO RF:R=I
14200 GOSUB 9500:A1=A1+VAL(D$)
14210 NEXT I
14220 A$=STR$(A1):L9=LEN(A$)-1
14225 IF A1<0 THEN RETURN
14230 A$=RIGHT$(A$,L9):RETURN
15000 REM *** COPIA ***
15010 X=1:Y=24:GOSUB11000:POKE 199,RV:PRINT"▶← : PARA SALIR▮";
15020 POKE199,NR:X=1:Y=22:GOSUB11000
15030 PRINT"LINEA INICIAL?";
15031 X=16:CC=0:C$=""
15032 GET A$:IF A$="" THEN 15032
15033 IF ASC(A$)<>95 AND (CC=0 AND (A$<"1"ORA$>"9")) THEN 15032
15034 IF ASC(A$)<>95 AND ASC(A$)<>13 AND (CC=1 AND (A$<"1"ORA$>"5")) THEN 15032
15035 IF ASC(A$)=95 THEN 15150
15036 IF ASC(A$)=13 AND CC=1 THEN 15039
15037 GOSUB11000:PRINTA$;:C$=C$+A$:X=X+1:CC=CC+1:IF CC=2 THEN 15039
15038 GOTO 15032
15039 RI=VAL(C$):YT=22:GOSUB 3000:IF RI>15 THEN 15020
15040 X=1:Y=22:GOSUB11000:PRINT"LINEA FINAL?";
15041 X=14:CC=0:C$=""
15042 GET A$:IF A$="" THEN 15042
15043 IF ASC(A$)<>95 AND (CC=0 AND (A$<"1"ORA$>"9")) THEN 15042
15044 IF ASC(A$)<>95 AND ASC(A$)<>13 AND (CC=1 AND (A$<"1"ORA$>"5")) THEN 15042
15045 IF ASC(A$)=95 THEN 15150
15046 IF ASC(A$)=13 AND CC=1 THEN 15050
15047 GOSUB11000:PRINTA$;:C$=C$+A$:X=X+1:CC=CC+1:IF CC=2 THEN 15050
15048 GOTO 15042
15050 RF=VAL(C$):IF RF>15 THEN 15040
15090 X=1:Y=22:GOSUB11000:PRINT"LINEA PRINCIPIO COPIA?";:X=24
15091 CC=0:C$=""
15092 GET A$:IF A$="" THEN 15092
15093 IF ASC(A$)<>95 AND (CC=0 AND (A$<"1"ORA$>"9")) THEN 15092
15094 IF ASC(A$)<>95 AND ASC(A$)<>13 AND (CC=1 AND (A$<"1"ORA$>"5")) THEN 15092
15095 IF ASC(A$)=95 THEN 15150
15096 IF ASC(A$)=13 AND CC=1 THEN 15100
15097 GOSUB11000:PRINTA$;:C$=C$+A$:X=X+1:CC=CC+1:IF CC=2 THEN 15100
15098 GOTO 15092
15100 RC=VAL(C$):IF RC>15 THEN 15090
15105 POKE646,5
15130 FOR I=RI TO RF:IF RC>15 THEN 15145
15132 B$(RC)=B$(I)
15140 RC=RC+1
15145 NEXT I:GOSUB8000
15150 YT=22:GOSUB3000:YT=YT+2:GOSUB3000:RETURN
16000 REM ***** COMANDOS *****
16005 I=0
16010 I=I+1:IF I=8 THEN 16060
16020 IF CO(I)=A0 THEN 16035
16030 GOTO 16010
16035 GOTO 16500
16040 ON I GOSUB 17500,18000,18500,20000,30000,30500,15000
16045 X=1:Y=24:GOSUB11000:PRINT"▮                    ";
16050 X=16:Y=24:GOSUB11000:POKE 199,NR:PRINT"▮COMANDOS : F A N I M L C▮";
16060 GOTO 5165
16500 REM ----- PRESENTA COMANDO -----
16510 X=27+(I-1)*2:Y=24:GOSUB11000
16520 POKE 199,RV
16530 PRINT "▮"MID$(V$,I,1)"▮";:POKE 199,NR:Y=21:GOSUB11000:GOTO 16040
17000 REM CALCULO
17010 PF=VAL(PF$):C$=F0$(PF):IF PF=0 OR F0$<>"2" THEN RETURN
17030 A$=F$(PF):RR=R:CC=C:GOSUB 9400
17035 L=LEN(A$):IFER<>0THENA$="ERR="+RIGHT$(STR$(ER),LEN(STR$(ER))-1):GOTO 17110
17040 IF L<9 THEN 17110
17050 IF VAL(A$)=INT(VAL(A$)) THEN A$="EEEEEEE":GOTO17110
17060 FOR I=L TO 1 STEP-1
17070 CH$=MID$(A$,I,1):IF CH$="." THEN 17090
17080 NEXT I:ERR=10:RETURN
17090 IF I>8 THEN A$="EEEEEEE":GOTO 17110
17100 A$=LEFT$(A$,7)
17110 R=RR:C=CC:GOSUB 9500:D$=A$
17120 GOSUB 9600:RETURN
17500 REM INTRUDUCCION FORMULA
17505 X=1:Y=24:GOSUB11000:POKE 199,RV:PRINT"▶← : PARA SALIR▮";
17510 FOR I=1 TO 8:IF F$(I)="" THEN 17530
17520 NEXT I
17530 PF=VAL(PF$):IF PF=0 THEN PF=I:PF$=RIGHT$(STR$(I),1)
17535 F0$="2"
17537 C$=F$(PF):X=1:Y=22:GOSUB11000:PRINT"▮FORM.:"
17550 L0=30:X=7:Y=21:Z$=C$:GOSUB2900:C$=Z$:GOSUB 3500:Z$=C$:GOSUB 2900
17555 IF Z$="" THEN PF$="0"
17556 GOSUB 9600
17560 F$(PF)=Z$:YT=22:GOSUB 3000:C$="":GOSUB 17000:RETURN
18000 REM INTRODUCCION ALFANUMERICOS
18002 X=1:Y=24:GOSUB11000:POKE 199,RV:PRINT"▶← : PARA SALIR▮";
18005 PF=VAL(PF$):F$(PF)=""
18010 PF$="0":F0$="1":X=1:Y=22:GOSUB11000:PRINT"▮DATO: ▮"
18030 L0=7:X=7:Y=21:Z$=D$:GOSUB2900:C$=Z$:GOSUB 3500
18040 D$=C$:GOSUB 9600
18050 YT=22:GOSUB 3000:YT=YT+2:GOSUB3000:RETURN
18500 REM INTRODUCCION NUMERICOS
18502 X=1:Y=24:GOSUB11000:POKE 199,RV:PRINT"▶← : PARA SALIR▮";
18510 F0$="2":X=1:Y=22:GOSUB11000:PRINT"▮NUM.: "
18520 CC=0:L0=7:X=7:Y=21:Z$=D$:GOSUB2900:C$=Z$:GOSUB 3500
18525 Z$=C$:GOSUB 2900:C$=Z$
18527 IF LEN(C$)=0 THEN 18550
18530 FOR I=1 TO LEN(C$):CH$=MID$(C$,I,1)
18535 IF (CH$<"0"ORCH$>"9")ANDCH$<>"."ANDCH$<>"-"THEN18520
18540 NEXT I:D$=C$:GOSUB 9600:GOSUB 1300
18550 YT=22:GOSUB 3000:YT=YT+2:GOSUB3000:RETURN
19000 REM * * *
19010 FOR I=15 TO R+1 STEP-1
19020 B$(I)=B$(I-1):NEXT I
19030 B$(R)="":FOR I=1 TO 6:B$(R)=B$(R)+BA$
19040 NEXT I:GOSUB 8100:RETURN
19500 RETURN
20000 REM *** **** ***
20010 X=1:Y=22:GOSUB11000:POKE 199,NR:PRINT"▮L C E";
20020 GOSUB 1200:IF A0<>76 AND A0<>67 AND A0<>69 THEN 20020
20030 Y=22:GOSUB11000:POKE 199,RV:IFA0=76THENX=1:GOSUB11000:PRINT"▮L▮";
20040 IF A0=67 THEN X=3:GOSUB11000:PRINT"▮C▮";
20050 IFA0=69 THEN 20080
20060 IF A0=76 THEN GOSUB19000:GOTO20080
20070 GOSUB 20500
20080 POKE 199,NR:YT=22:GOSUB 3000:RETURN
20500 REM * * *
20510 K=C:FOR IC=25 TO K STEP-1
20520 C=IC:GOSUB 9500:C=IC+1
20530 GOSUB 9600:NEXT IC
20540 PF$="0":F0$="1":D$="        ":C=K
20550 GOSUB 9600:GOSUB 8100:RETURN
30000 REM ** DISCO **
30010 X=1:Y=24:GOSUB11000:POKE 199,RV:PRINT"▶← : PARA SALIR▮";
30011 POKE199,NR:X=1:Y=22:GOSUB11000
30012 PRINT"▮CELDA INICIAL?";
30014 X=17:CC=0:C$=""
30016 GET A$:IF A$="" THEN 30016
30018 IF ASC(A$)<>95 AND (CC=0 AND (A$<"A"ORA$>"O")) THEN 30016
30020 IF ASC(A$)<>95 AND (CC=1 AND (A$<"1"ORA$>"9")) THEN 30016
30021 IF ASC(A$)<>95 AND ASC(A$)<>13 AND (CC=2 AND (A$<"0"ORA$>"6")) THEN 30016
30022 IF ASC(A$)=95 THEN 30260
30024 IF ASC(A$)=13 AND CC=2 THEN 30029
30026 GOSUB11000:PRINTA$;:C$=C$+A$:X=X+1:CC=CC+1:IF CC=3 THEN 30029
30028 GOTO 30016
30029 RI$=LEFT$(C$,1):CI$=MID$(C$,2,LEN(C$)-1):YT=22:GOSUB 3000
30030 IF VAL(CI$)>26 THEN 30011
30035 POKE199,NR:X=1:Y=22:GOSUB11000
30036 PRINT"CELDA FINAL?";
30037 X=15:CC=0:C$=""
30040 GET A$:IF A$="" THEN 30040
30042 IF ASC(A$)<>95 AND (CC=0 AND (A$<"A"ORA$>"O")) THEN 30040
30044 IF ASC(A$)<>95 AND (CC=1 AND (A$<"1"ORA$>"9")) THEN 30040
30045 IF ASC(A$)<>95 AND ASC(A$)<>13 AND (CC=2 AND (A$<"0"ORA$>"6")) THEN 30040
30046 IF ASC(A$)=95 THEN 30260
30048 IF ASC(A$)=13 AND CC=2 THEN 30054
30050 GOSUB11000:PRINTA$;:C$=C$+A$:X=X+1:CC=CC+1:IF CC=3 THEN 30054
30052 GOTO 30040
30054 RF$=LEFT$(C$,1):CF$=MID$(C$,2,LEN(C$)-1)
30055 X=1:Y=22:GOSUB11000:PRINT"MEMORIZACION";LEFT$(BA$,10);:YT=24:GOSUB3000
30056 IF VAL(CF$)>26 THEN 30011
30060 CI=0:CF=0:RI=0:RF=0
30090 FOR I=1 TO 15:IF MID$(R$,I,1)=RI$ THEN RI=I
30100 IF MID$(R$,I,1)=RF$ THEN RF=I
30110 NEXT I
30120 FOR I=1 TO 26:IF VAL(CI$)=I THEN CI=I
30130 IF VAL(CF$)=I THEN CF=I
30140 NEXT I:IF CF=0 OR RF=0 THEN GOSUB 3000:GOTO 30010
30142 FOR I=1 TO 9:F$(I)=F$(I)+LEFT$(BA$,30-LEN(F$(I))):NEXT I
```

```
30143 POKE 646,5
30145 GOSUB 6500
30150 GOSUB 6000
30152 FOR I=1 TO 15:AA$="":ED=0
30155 OPEN2,8,3,"TTBB":GOSUB 6004:IF ED<>0 THEN 30260
30160 FOR J=1 TO 26
30180 IF I<RI OR J<CI OR I>RF OR J>CF THEN AA$=AA$+"#########":GOTO 30202
30190 R=I:C=J:GOSUB 9500:IF F0$=" "THEN F0$="#"
30200 AA$=AA$+F0$+PF$+D$
30202 NEXT J
30204 PA=I:GOSUB 6800:PRINT#2,AA$
30205 CLOSE2:GOSUB 6004:IF ED<>0 THEN 30260
30220 NEXT I
30230 FOR I=1 TO 9:GOSUB30602:AA$=F$(I):ED=0
30235 OPEN2,8,3,"TTBB"
30240 PA=15+I:GOSUB 6800:PRINT#2,AA$:CLOSE2:GOSUB 6004
30242 IF ED<>0 THEN 30260
30245 NEXT I
30250 CLOSE1:CLOSE2
30260 YT=22:GOSUB3000:YT=YT+2:GOSUB3000
30270 GOSUB 8000:RETURN
30500 REM ** CARGA **
30510 ED=0:GOSUB 6000:OPEN2,8,3,"TTBB":LD=1:GOSUB 6004
30515 IF ED<>0 THEN 30600
30520 FOR I=1 TO 15:PA=I:GOSUB 6800
30525 AA$="":FOR K1=1 TO 234:GET#2,A$:AA$=AA$+A$:NEXT K1
30530 B$(I)=AA$:GOSUB 30700:NEXT I
30540 FOR IC=1 TO 9:PA=IC+15
30542 GOSUB 6800
30545 INPUT#2,AA$
30550 FOR K9=1 TO LEN(AA$):CH$=MID$(AA$,K9,1)
30552 IF CH$="#" THEN CH$=CHR$(32)
30555 AA$=LEFT$(AA$,K9-1)+CH$+RIGHT$(AA$,LEN(AA$)-K9)
30557 NEXT K9
30560 Z$=AA$:GOSUB 2900:F$(IC)=Z$
30570 NEXT IC:CLOSE1:CLOSE2
30600 GOSUB 8000:LD=0:RETURN
30602 REM *** SUSTITUYE ESPACIOS ***
30605 FOR G9=1 TO LEN(F$(I))
30607 CH$=MID$(F$(I),G9,1):IF CH$=CHR$(32) THEN CH$="#"
30610 F$(I)=LEFT$(F$(I),G9-1)+CH$+RIGHT$(F$(I),LEN(F$(I))-G9)
30615 NEXT G9:RETURN
30700 REM *** SUSTITUYE # ***
30705 FOR K9=1 TO 234:G$=MID$(B$(I),K9,1)
30710 IF G$="#" THEN G$=CHR$(32)
30715 B$(I)=LEFT$(B$(I),K9-1)+G$+RIGHT$(B$(I),234-K9)
30720 NEXT K9:RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES COMMODORE 64

OP$(*)	= matriz con los caracteres de reconocimiento operador (↑,/,*,–,+)
FZ$(*)	= matriz con las funciones reconocidas (ABS,INT,COS,SIN,SQR,TAN)
CO(*)	= códigos de comandos
B$(*)	= matriz con las 15 líneas del tablero
HO$	= borrado pantalla
BA$,BU$ ZE$,F1$	= cadenas auxiliares para procesos y preparaciones (subrutina 500)
RV	= flag para leyendas invertidas
NR	= flag para leyendas normales
A$	= tecla pulsada (rutina 1200)
A0	= código ASCII de A$
DD$	= cadena para presentación celda
XX,YY	= coordenadas en pantalla de celda
PF$	= puntero a fórmula
C$	= cadena para introducción datos (subrutina 3500)
R	= línea actual
C	= columna actual
AD,X$ CD,DD	= indicadores del estado de la unidad disco
LD	= flag para el comando LOAD
PA	= registro actual
ER	= indicador de error
F0$	= indicador tipo de celda (1 = alfanumérico; 2 = numérico)
F$(*)	= matriz para fórmulas



```
1     REM    --------------------
2     REM    TABLERO ELECTRONICO
3     REM       VERSION   DOS
4     REM    --------------------
10    DATA "^","/","*","-","+"
20    DATA "ABS","INT","COS","SIN","SQR
      ","TAN"
30    FOR I = 1 TO 5:
         READ OP$(I):
      NEXT
35    DATA 70,65,78,73,77,76,67
40    FOR I = 1 TO 6:
         READ FZ$(I):
      NEXT
50    FOR I = 1 TO 7:
         READ CO(I):
      NEXT
60    GOSUB 500:
      GOTO 5000
70    STOP
500   REM    ---------------------------
530   TEXT :
      HOME :
      BP$ =  CHR$ (7):
      DR$ = "1":
      D1$ =  CHR$ (4):
      B$ = "          ";
      BA$ = B$ + B$ + B$ + B$:
      B$ = "__________";
      BU$ = B$ + B$ + B$ + B$:
      B$ = ""
540   DIM B$(15)
550   FOR I = 1 TO 15:
         ZE$ = ZE$ + "0000":
         FOR J = 1 TO 24:
            B$(I) = B$(I) +  LEFT$ (BA$
            ,10):
      NEXT J,I
660   FC = 1:
      RETURN
1000  REM --- ENCIENDE CURSOR ---
1010  X1 = X - 1:
      Y1 = Y - 1:
      CH =  SCRN( X1,2 * Y1) + 16
       *  SCRN( X1,2 * Y1 + 1):
      HTAB X:
      VTAB Y:
      INVERSE
1040  PRINT  CHR$ (CH);:
      HTAB X:
      NORMAL :
      RETURN
1100  REM --- APAGA CURSOR ---
1110  X = X1 + 1:
      Y = Y1 + 1:
      NORMAL :
      HTAB X:
      VTAB Y:
      PRINT  CHR$ (CH);:
      HTAB X:
      RETURN
1200  REM
1210  A0 =  PEEK ( - 16384):
      IF A0 > 128 THEN  POKE  - 16368,0
      :
      A0 = A0 - 128:
      RETURN
1220  GOTO 1210
1300  REM --- ENCIENDE CELDA ---
1310  NORMAL :
      HTAB XX:
      VTAB YY:
      PRINT DD$
1320  XX = DX * 8 + 2:
      YY = DY * 2 + 2
1330  GOSUB 9500:
      INVERSE :
      IF PF$ <  > "0" AND PF$ <
       > " " THEN  GOSUB 17000
1340  HTAB XX:
      VTAB YY:
      PRINT D$:
      DD$ = D$
1350  NORMAL :
      RETURN
2700  REM --- CENTRA EN PANTALLA ---
2730  XT = (42 -  LEN (TI$)) / 2:
      HTAB XT:
      VTAB YT:
      PRINT TI$;:
      RETURN
2900  REM --- ELIMINA ESPACIOS ---
2930  Z2$ = "":
      A =  LEN (Z$):
      FOR I = 1 TO A:
         Z1$ =  MID$ (Z$,I,1):
         IF Z1$ <  > " " THEN Z2$
          = Z2$ + Z1$
2950  NEXT I:
      Z$ = Z2$:
      RETURN
3000  REM --- BORRA LINEA YT ---
3010  HTAB 1:
      VTAB YT:
      PRINT  LEFT$ (BA$,39);
3020  RETURN
3500  REM --- ENTRADA CAMPO ---
3530  IC = 0:
      C$ = B$(CC):
      IF LO = 0 THEN  RETURN
3535  IF CC = 0 THEN  HTAB XC:
      VTAB YC:
      PRINT  MID$ (BU$,1,LO)
3537  HTAB XC:
      VTAB YC:
      PRINT C$;
3540  X = XC:
      Y = YC
3545  IF  MID$ (C$,IC + 1,1) = "."
       AND TP(CC) = 1 THEN IC = IC
       + 1:
      X = X + 1
3550  GOSUB 1000:
      GOSUB 1200:
      GOSUB 1100
3560  IF A0 = 5 OR A0 = 23 OR A0
       = 26 OR A0 = 18 OR A0 = 27
       THEN  RETURN
3580  IF A0 <  > 13 THEN 3620
3582  IF IC = 0 THEN  RETURN
3585  IF TP(CC) = 1 AND  MID$ (C$,IC,1)
       = "." THEN IC = IC - 1
3590  C$ =  MID$ (C$,1,IC):
      A$ =  MID$ (BA$,1,LO - IC):
      C$ = C$ + A$:
      A$ =  MID$ (BU$,1,LO - IC):
      IF TP(CC) = 1 THEN A$ =  MID$ (BE
      $,1,LO - IC)
3605  HTAB XC + IC:
      PRINT A$:
      RETURN
3620  IF A0 <  > 1 THEN 3650
3630  IF IC = 0 THEN  PRINT BP$;:
      GOTO 3550
3640  IC = IC - 1:
      X = X - 1:
      GOTO 3550
3650  IF IC = LO THEN  PRINT BP$;:
      GOTO 3550
3655  IF A0 = 0 THEN 3550
3660  IF A0 = 19 THEN IC = IC + 1:
      X = X + 1:
      GOTO 3545
3670  CH$ =  CHR$ (A0):
      PRINT CH$;:
      C$ =  MID$ (C$,1,IC) + CH$
       +  MID$ (C$,IC + 2,39):
      IC = IC + 1:
      X = X + 1:
      GOTO 3545
5000  REM --- MAIN ---
5010  GOSUB 8000:
      DD$ = "":
      XX = 1:
      YY = 1
5020  DX = C - C1:
      DY = R - R1
5030  GOSUB 1300:
      GOSUB 1200
5040  IF A0 <  > 19 THEN 5070
5050  IF DX = 4 AND C1 < 22 THEN C
       = C + 1:
      C1 = C1 + 1:
      GOSUB 8100:
      GOTO 5020
5060  IF DX < 4 THEN C = C + 1:
      GOTO 5020
5070  IF A0 <  > 1 THEN 5100
5080  IF DX = 0 AND C1 > 1 THEN C
       = C - 1:
      C1 = C1 - 1:
      GOSUB 8100:
      GOTO 5020
5090  IF DX > 0 THEN C = C - 1:
      GOTO 5020
5100  IF A0 <  > 26 THEN 5130
5110  IF DY = 9 AND R1 < 6 THEN R
       = R + 1:
      R1 = R1 + 1:
      GOSUB 8100:
      GOTO 5020
5120  IF DY < 9 THEN R = R + 1:
      GOTO 5020
5130  IF A0 <  > 23 THEN 5160
5140  IF DY = 0 AND R1 > 1 THEN R
       = R - 1:
      R1 = R1 - 1:
      GOSUB 8100:
      GOTO 5020
5150  IF DY > 0 THEN R = R - 1:
      GOTO 5020
5160  GOTO 16000
6000  REM --- OPEN ---
6030  PRINT D1$
6040  PRINT D1$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D1$
6060  RETURN
6200  REM --- GET ---
6230  PRINT D1$
6240  PRINT D1$;"READ";NF$;",R";NR
6250  INPUT AA$
6260  PRINT D1$
```

```
6270 RETURN
6400 REM  --- PUT ---
6430 PRINT D1$
6440 PRINT D1$;"WRITE";NF$;",R";NR
6450 PRINT  CHR$ (34) + AA$
6460 PRINT D1$
6470 RETURN
6600 REM  --- CLOSE ---
6630 PRINT D1$
6640 PRINT D1$;"CLOSE";NF$
6650 PRINT D1$
6660 RETURN
8000 REM  --- INICIALIZA PANTALLA ---
8010 R$ = "":
     FOR I = 65 TO 79:
        R$ = R$ +  CHR$ (I):
     NEXT I:
     INVERSE
8015 V$ = "FANIMLC"
8020 R1 = 1:
     C1 = 1:
     R = 1:
     C = 1:
     FOR I = 2 TO 20:
        HTAB 1:
        VTAB I:
        PRINT " ";:
     NEXT I
8025 NORMAL :
     HTAB 27:
     VTAB 24:
     PRINT "F A N I M L C";
8100 REM  --- SCROLL ---
8110 INVERSE :
     C2 = C1:
     HTAB 1:
     VTAB 1:
     PRINT  LEFT$ (BA$,40);
8120 VTAB 1:
     FOR I = 5 TO 37 STEP 8
8130    HTAB I:
        PRINT C2;:
        C2 = C2 + 1:
     NEXT I
8140 R2 = R1:
     FOR I = 2 TO 20 STEP 2
8150    HTAB 1:
        VTAB I:
        PRINT  MID$ (R$,R2,1):
        R2 = R2 + 1
8160 NEXT I:
     NORMAL
8170 Y = 2:
     FOR I = R1 TO R1 + 9
8180    X = 2:
        FOR J = C1 TO C1 + 4
8190       N1 = 1 + 9 * (J - 1):
           A$ =  MID$ (B$(I),N1,9)
8195       FO$ =  LEFT$ (A$,1):
           PF$ =  MID$ (A$,2,1)
8200       D$ =  RIGHT$ (A$,7):
           HTAB X:
           VTAB Y
8210       PRINT D$:
           X = X + 8:
        NEXT J
8220    Y = Y + 2:
     NEXT I:
     RETURN
9000 REM  --- CAL 2 ---
9005 ER = 0:
     L =  LEN (CA$):
     IF L < 3 THEN 9122
9010 FOR I = 2 TO L - 1:
        CH$ =  MID$ (CA$,I,1):
        FOR J = 1 TO 5:
           IF OP$(J) = CH$ THEN 9020
9015 NEXT J,I:
     ER = 1:
     GOTO 9122
9020 CH$ =  MID$ (CA$,I + 1,1):
     IF CH$ = "*" OR CH$ = "^" OR CH$
      = "/" THEN ER = 2:
     GOTO 9122
9025 N1 =  VAL ( MID$ (CA$,1,I - 1)):
     N2 =  VAL ( MID$ (CA$,I + 1,40)):
     ON J GOTO 9035,9040,9050,9055,906
     0:
     CA$ =  STR$ (N1):
     IF N1 >  = 0 THEN CA$ = "+"
      + CA$
9027 CA$ =  STR$ (N1):
     IF N1 >  = 0 THEN CA$ = "+"
      + CA$
9030 GOTO 9122
9035 N1 = N1 ^ N2:
     GOTO 9027
9040 IF N2 = 0 THEN ER = 3:
     GOTO 9027
9045 N1 = N1 / N2:
     GOTO 9027
9050 N1 = N1 * N2:
     GOTO 9027
9055 N1 = N1 - N2:
     GOTO 9027
9060 N1 = N1 + N2:
     GOTO 9027
9065 REM  --- FORM 2 ---
9070 ER = 0:
     LO =  LEN (FO$):
     IF LO >  = 3 THEN 9075
9072 IF EG = 1 THEN 9192
9074 GOTO 9162
9075 FOR OP = 1 TO 5:
        PO = 0:
        P1 = 0:
        P2 = 0:
        FOR IO = 2 TO LO - 1:
           CH$ =  MID$ (FO$,IO,1):
           CH =  ASC (CH$):
           IF CH$ = OP$(OP) AND PO
            = 0 THEN PO = IO:
           GOTO 9095
9080       O = 0:
           IF (CH = 94 OR CH < 48)
            AND CH <  > 46 THEN O
            = 1
9085       IF O = 1 AND PO = 0 THEN P1
            = IO
9090       IF O = 1 AND PO > 0 AND IO
            - PO > 1 THEN P2 = IO:
           GOTO 9105
9095    NEXT IO:
        IF PO > 0 THEN 9105
9100 NEXT OP:
     IF EG = 1 THEN 9192
9102 GOTO 9162
9105 IF P1 = 0 THEN P1 = 1:
     GOTO 9115
9110 CH$ =  MID$ (FO$,P1,1):
     IF CH$ <  > "+" AND CH$ <
      > "-" THEN P1 = P1 + 1
9115 P2 = P2 - 1:
     IF P2 < 0 THEN P2 = LO
9120 CA$ =  MID$ (FO$,P1,P2 - P1
      + 1):
     GOTO 9000
9122 IF ER = 0 THEN 9125
9123 IF EG = 1 THEN 9192
9124 GOTO 9162
9125 GOTO 9280
9127 FO$ =  MID$ (FO$,1,P1 - 1)
      + CA$ +  MID$ (FO$,P2 + 1,40):
     GOTO 9070
9130 REM  --- PAR 1 ---
9140 ER = 0:
     LA =  LEN (A$):
     IF LA < 3 THEN 9165
9145 PI = 0:
     PF = 0:
     FOR IA = 1 TO LA:
        CH$ =  MID$ (A$,IA,1):
        IF CH$ = "(" THEN PI = IA
9150    IF CH$ = ")" THEN 9185
9155 NEXT IA:
     IF PI > 0 THEN ER = 4:
     GOTO 17035
9160 FO$ = A$:
     GOTO 9065
9162 A$ = FO$
9165 IF  VAL (A$) = 0 THEN A$ = "0":
     GOTO 17035
9175 IF  LEFT$ (A$,1) = "+" THEN A$
      =  MID$ (A$,2,80)
9180 GOTO 17035
9185 IF PI = 0 THEN ER = 4:
     GOTO 17035
9190 PF = IA:
     FO$ =  MID$ (A$,PI + 1,PF - PI
      - 1):
     EG = 1:
     GOTO 9065
9192 EG = 0:
     IF ER <  > 0 THEN 17035
9195 IF PI > 3 THEN  GOSUB 9220
9200 IF PI <  = 1 THEN 9210
9205 IF A < 0 AND  MID$ (A$,PI - 1,1)
      = "-" THEN PI = PI - 1:
     FO$ = "+" +  MID$ (FO$,2,80)
9210 IF ER <  > 0 THEN 17035
9215 A$ =  MID$ (A$,1,PI - 1) + FO$
      +  MID$ (A$,PF + 1,80):
     GOTO 9140
9220 REM  --- FUN 1 ---
9225 ER = 0:
     CH$ =  MID$ (A$,PI - 1,1):
     IF CH$ < "A" OR CH$ > "Z" THEN
      RETURN
9230 FU$ =  MID$ (A$,PI - 3,3):
     FOR I = 1 TO 6:
        IF FU$ = FZ$(I) THEN 9240
9235 NEXT I:
     ER = 5:
     RETURN
9240 A1 =  VAL (FO$):
     ON I GOSUB 9245,9250,9255,9260,92
     65,9275:
     CA$ =  STR$ (A1):
     IF A1 >  = 0 THEN CA$ = "+"
      + CA$
9241 EP = 1:
     GOTO 9280
9242 FO$ = CA$:
     PI = PI - 3:
     RETURN
9245 A1 =  ABS (A1):
     RETURN
9250 A1 =  INT (A1):
     RETURN
9255 A1 =  COS (A1):
     RETURN
9260 A1 =  SIN (A1):
     RETURN
9265 IF A1 < 0 THEN ER = 6:
     RETURN
9270 A1 =  SQR (A1):
     RETURN
9275 A1 =  TAN (A1):
     RETURN
9280 REM  --- ESP 2 ---
9290 L =  LEN (CA$):
     FOR IC = 1 TO L:
        IF  MID$ (CA$,IC,1) = "E"
         THEN 9300
9295 NEXT :
     IF EP = 1 THEN EP = 0:
     GOTO 9242
```

```
9297 GOTO 9127
9300 N$ =  MID$ (CA$,2,IC - 2):
     L =  LEN (N$)
9305 FOR I = 1 TO L:
        IF  MID$ (N$,I,1) = "."
        THEN P = I:
        GOTO 9315
9310 NEXT :
     P = L
9315 E =  VAL ( MID$ (CA$,IC + 1,3)):
     IF E < 0 THEN 9355
9320 DD = L - P:
     IF DD = 0 THEN 9345
9325 N$ =  MID$ (N$,1,P - 1) +
      MID$ (N$,P + 1,80)
9330 IF DD <  = E THEN 9345
9335 N$ =  MID$ (N$,1,P + E - 1)
      + "." +  MID$ (N$,P + E,80):
     DD = E
9345 N$ = N$ +  MID$ (ZE$,1,E - DD):
     GOTO 9375
9355 IF P = L THEN P = P + 2 - L:
     GOTO 9365
9360 N$ =  MID$ (N$,1,P - 1) +
      MID$ (N$,P + 1,80):
     L = 1
9365 N$ =  MID$ (ZE$,1, - E - L
      + 1) + N$
9370 N$ =  MID$ (N$,1,P - 1) + "."
      +  MID$ (N$,P,80)
9375 CA$ =  LEFT$ (CA$,1) + N$:
     IF EP = 1 THEN EP = 0:
     GOTO 9242
9377 GOTO 9127
9400 REM  --- VAR 1 ---
9405 Z$ = A$:
     GOSUB 2900:
     A$ = Z$
9410 LA =  LEN (A$):
     IF  LEFT$ (A$,3) = "SUM" THEN
      GOSUB 14000:
     GOTO 17035
9415 FOR I = 1 TO LA
9420    CH$ =  MID$ (A$,I,1):
        IF CH$ >  = "A" AND CH$
         <  = "Z" THEN 9440
9430 NEXT I:
     GOTO 9130
9440 PI = I:
     C$ =  MID$ (A$,I + 2,1):
     IF C$ >  = "A" AND C$ <  = "Z"
      THEN I = I + 2:
     GOTO 9430
9450 IF CH$ > "O" THEN ER = 7:
     RETURN
9460 R =  ASC (CH$) - 64:
     C =  VAL ( MID$ (A$,PI + 1,1)):
     IF C = 0 THEN ER = 7:
     RETURN
9470 GOSUB 9500:
     Z$ = D$:
     GOSUB 2900
9480 A$ =  MID$ (A$,1,PI - 1) + Z$
      +  MID$ (A$,PI + 2,80)
9490 GOTO 9410
9500 REM  --- LECTURA CELDA ---
9510 N1 = 1 + 9 * (C - 1):
     B$ =  MID$ (B$(R),N1,9)
9520 FO$ =  LEFT$ (B$,1):
     PF$ =  MID$ (B$,2,1)
9530 D$ =  RIGHT$ (B$,7):
     RETURN
9600 REM  --- ESCRITURA CELDA ---
9610 N1 = 9 * (C - 1):
     N2 = N1 + 10
9620 A1$ =  MID$ (B$(R),1,N1):
     A2$ =  MID$ (B$(R),N2,235)
9630 D$ =  LEFT$ (D$,7)
9640 IF FO$ = "1" THEN D$ = D$ +
      MID$ (BA$,1,7 -  LEN (D$))
9650 IF FO$ = "2" THEN D$ =  MID$ (BA$
     ,1,7 -  LEN (D$)) + D$
9660 B$(R) = A1$ + FO$ + PF$ + D$
      + A2$:
     RETURN
14000 REM --- SUM ---
14010 ER = 0:
      FOR I = 5 TO LA - 1
14020    IF  MID$ (A$,I,1) = "."
          THEN 14040
14030 NEXT I:
      ER = 10:
      A$ = "ERR=10":
      RETURN
14040 A1$ =  MID$ (A$,5,I - 5):
      A2$ =  MID$ (A$,I + 1,LA - I
       - 1)
14060 RI$ =  LEFT$ (A1$,1):
      RF$ =  LEFT$ (A2$,1)
14080 CI =  VAL ( MID$ (A1$,2,2)):
      CF =  VAL ( MID$ (A2$,2,2))
14090 RI = 0:
      RF = 0:
      A1 = 0
14100 FOR I = 1 TO 15:
         CH$ =  MID$ (R$,I,1)
14110    IF CH$ = RI$ THEN RI = I
14120    IF CH$ = RF$ THEN RF = I
14130 NEXT I
14140 IF RI = 0 OR RF = 0 OR CI = 0
       OR CF = 0 THEN ER = 10:
      A$ = "ERR=10":
      RETURN
14145 IF RI <  > RF AND CI <  > CF
       THEN RI = 0:
      GOTO 14140
14150 IF CI = CF THEN 14190
14160 R = RI:
      FOR I = CI TO CF:
         C = I
14170    GOSUB 9500:
         A1 = A1 +  VAL (D$)
14180 NEXT I:
      GOTO 14220
14190 C = CI:
      FOR I = RI TO RF:
         R = I
14200    GOSUB 9500:
         A1 = A1 +  VAL (D$)
14210 NEXT I
14220 A$ =  STR$ (A1):
      RETURN
15000 REM --- COPIA ---
15010 HTAB 1:
      VTAB 24:
      INVERSE :
      PRINT "(ESC PARA SALIR)";
15020 NORMAL :
      HTAB 1:
      VTAB 22:
      LO = 2
15030 PRINT "LINEA INICIAL ?":
      XC = 17:
      YC = 22
15040 B$(0) = "":
      CC = 0:
      GOSUB 3500:
      IF AO = 27 THEN 15150
15050 RI =  VAL (C$):
      YT = 22:
      GOSUB 3000
15060 HTAB 1:
      VTAB 22:
      PRINT "LINEA FINAL ?"
15070 XC = 15:
      GOSUB 3500:
      IF AO = 27 THEN 15150
15080 RF =  VAL (C$):
      HTAB 1:
      VTAB 22
15090 PRINT "LINEA PRINC. COPIA?":
      XC = 20
15100 GOSUB 3500:
      IF AO = 27 THEN 15150
15110 RC =  VAL (C$):
      GOSUB 3000
15120 IF RI <  = 0 OR RF <  = 0 OR RC
       <  = 0 OR RI > 15 OR RF > 15
       OR RC > 15 OR RI > RF THEN 15020
15130 FOR I = RI TO RF:
         IF RC > 15 THEN 15145
15140    B$(RC) = B$(I):
         RC = RC + 1
15145 NEXT I:
      GOSUB 8000
15150 GOSUB 3000:
      YT = 24:
      GOSUB 3000:
      RETURN
16000 REM --- COMANDOS ---
16010 FOR I = 1 TO 9
16020    IF CO(I) = AO THEN 16035
16030 NEXT I:
      GOTO 5020
16035 GOTO 16500
16040 ON I GOSUB 17500,18000,18500,2000
      0,30000,30500,15000
16050 HTAB 27:
      VTAB 24:
      NORMAL :
      PRINT "F A N I M L C";
16060 GOTO 5020
16500 REM --- PRESENTA COMANDOS ---
16510 X = 27 + (I - 1) * 2:
      HTAB X:
      VTAB 24
16520 FLASH
16530 PRINT  MID$ (V$,I,1);:
      NORMAL :
      VTAB 21:
      GOTO 16040
17000 REM --- CALCULO ---
17010 PF =  VAL (PF$):
      IF PF = 0 OR FO$ <  > "2" THEN
       RETURN
17030 A$ = F$(PF):
      RR = R:
      CC = C:
      GOTO 9400
17035 L =  LEN (A$):
      IF ER <  > 0 THEN A$ = "ERR="
       +  STR$ (ER):
      GOTO 17110
17040 IF L < 8 THEN 17110
17050 IF  VAL (A$) =  INT ( VAL (A$))
       THEN A$ = "EEEEEEE":
      GOTO 17110
17060 FOR I = L TO 1 STEP  - 1
17070    CH$ =  MID$ (A$,I,1):
         IF CH$ = "." THEN 17090
17080 NEXT I:
      ERR = 10:
      RETURN
17090 IF I > 8 THEN A$ = "EEEEEEE":
      GOTO 17110
17100 A$ =  LEFT$ (A$,7)
17110 R = RR:
      C = CC:
      GOSUB 9500:
      D$ = A$
17120 GOSUB 9600:
      RETURN
17500 REM --- INTRODUCCION FORMULA ---
17510 FOR I = 1 TO 8:
         IF F$(I) = "" THEN 17530
17520 NEXT I
```

```
17530 PF =  VAL (PF$):
      IF PF = 0 THEN PF = I:
      PF$ =  STR$ (I)
17535 FO$ = "2"
17537 B$(0) = F$(PF):
      CC = 0
17540 HTAB 1:
      VTAB 22:
      PRINT "FORM. "
17550 LO = 30:
      XC = 8:
      YC = 22:
      CC = 0:
      GOSUB 3500:
      Z$ = C$:
      GOSUB 2900
17555 IF Z$ = "" THEN PF$ ="0"
17556 GOSUB 9600
17560 F$(PF) = Z$:
      YT = 22:
      GOSUB 3000:
      B$(0) = "":
      GOSUB 17000:
      RETURN
18000 REM --- INTROD. ALFANUMERICA ---
18005 PF =  VAL (PF$):
      F$(PF) = ""
18010 PF$ = "0":
      FO$ = "1":
      CC = 0:
      Z$ = D$:
      GOSUB 2900:
      B$(0) = Z$
18020 HTAB 1:
      VTAB 22:
      PRINT "DATO. "
18030 LO = 7:
      XC = 7:
      YC = 22:
      GOSUB 3500
18040 D$ = C$:
      GOSUB 9600
18050 YT = 22:
      GOSUB 3000:
      RETURN
18500 REM --- INTROD. NUMERICA ---
18510 FO$ = "2":
      HTAB 1:
      VTAB 22:
      PRINT "DATO. "
18520 CC = 0:
      Z$ = D$:
      GOSUB 2900:
      B$(0) = Z$:
      LO = 7:
      XC = 7:
      YC = 22:
      GOSUB 3500
18525 Z$ = C$:
      GOSUB 2900:
      C$ = Z$
18530 FOR I = 1 TO  LEN (C$):
        CH$ =  MID$ (C$,I,1)
18535   IF (CH$ < "0" OR CH$ > "9")
         AND CH$ <  > "." AND CH$
         <  > "-" THEN  PRINT BP$;:
        GOTO 18520
18540 NEXT I:
      D$ = C$:
      GOSUB 9600:
      GOSUB 1300
18550 YT = 22:
      GOSUB 3000:
      RETURN
19000 REM --- INSER. LINEAS ---
19010 FOR I = 15 TO R + 1 STEP  - 1
19020   B$(I) = B$(I - 1):
      NEXT I
19030 B$(R) = "":
      FOR I = 1 TO 6:
        B$(R) = B$(R) + BA$
19040 NEXT I:
      GOSUB 8100:
      RETURN
19500 RETURN
20000 REM --- INSER. L y C ---
20010 HTAB 1:
      VTAB 24:
      NORMAL :
      PRINT "L C E";
20020 GOSUB 1200:
      IF AO <  > 76 AND AO <  > 67
       AND AO <  > 69 THEN 20020
20030 VTAB 24:
      INVERSE :
      IF AO = 76 THEN  HTAB 1:
      PRINT "L";
20040 IF AO = 67 THEN  HTAB 3:
      PRINT "C";
20050 IF AO = 69 THEN 20080
20060 IF AO = 76 THEN  GOSUB 19000:
      GOTO 20080
20070 GOSUB 20500
20080 NORMAL :
      YT = 24:
      GOSUB 3000:
      RETURN
20500 REM --- INSER. COLUMNAS ---
20510 K = C:
      FOR IC = 25 TO K STEP  - 1
20520   C = IC:
        GOSUB 9500:
        C = IC + 1
20530   GOSUB 9600:
      NEXT IC
20540 PF$ = "0":
      FO$ = "1":
      D$ = "         ":
      C = K
20550 GOSUB 9600:
      GOSUB 8100:
      RETURN
30000 REM --- DISCO ---
30010 HTAB 1:
      VTAB 24:
      INVERSE :
      PRINT "(ESC PARA SALIR)";
30020 NORMAL :
      HTAB 1:
      VTAB 22:
      LO = 3
30030 PRINT "CELDA INICIAL ?":
      XC = 17:
      YC = 22
30040 B$(0) = "":
      CC = 0:
      GOSUB 3500:
      IF AO = 27 THEN 30260
30045 RI$ =  LEFT$ (C$,1):
      CI$ =  RIGHT$ (C$,2)
30050 YT = 22:
      GOSUB 3000:
      HTAB 1:
      VTAB 22
30060 PRINT "CELDA FINAL ?":
      XC = 15
30070 GOSUB 3500:
      IF AO = 27 THEN 30260
30080 RF$ =  LEFT$ (C$,1):
      CF$ =  RIGHT$ (C$,2)
30085 CI = 0:
      CF = 0:
      RI = 0:
      RF = 0
30090 FOR I = 1 TO 15:
        IF  MID$ (R$,I,1) = RI$
         THEN RI = I
30100   IF  MID$ (R$,I,1) = RF$
         THEN RF = I
30110 NEXT I
30120 FOR I = 1 TO 26:
        IF  VAL (CI$) = I THEN CI
         = I
30130   IF  VAL (CF$) = I THEN CF
         = I
30140 NEXT I:
      IF CF = 0 OR CI = 0 OR RI = 0
       OR RF = 0 THEN  GOSUB 3000:
      GOTO 30010
30150 NF$ = "TTBB":
      LR = 234:
      GOSUB 6000
30160 FOR I = 1 TO 15:
        AA$ = ""
30170   FOR J = 1 TO 26
30180     IF I < RI OR J < CI OR I
           > (RF) OR J > CF THEN AA$
           = AA$ + "          ":
          GOTO 30210
30190     R = I:
          C = J:
          GOSUB 9500
30200     AA$ = AA$ + FO$ + PF$
           + D$
30210   NEXT J:
        NR = I:
        GOSUB 6400
30220 NEXT I
30230 FOR I = 1 TO 9:
        AA$ = F$(I)
30240   NR = 15 + I:
        GOSUB 6400:
      NEXT I
30250 GOSUB 6600
30260 YT = 24:
      GOSUB 3000:
      YT = 22:
      GOSUB 3000
30270 GOSUB 8000:
      RETURN
30500 REM --- LOAD ---
30510 NF$ = "TTBB":
      LR = 234:
      GOSUB 6000
30520 FOR I = 1 TO 15:
        NR = I:
        GOSUB 6200
30530   B$(I) = AA$:
      NEXT I
30540 FOR IC = 1 TO 9:
        NR = IC + 15:
        GOSUB 6200
30550   Z$ = AA$:
        GOSUB 2900:
        F$(IC) = Z$
30560 NEXT IC:
      GOSUB 6600
30570 GOSUB 8000:
      RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - APPLE

B$(*)	= matriz de las líneas para el tablero
OP$(*)	= matriz con los símbolos de los operadores (↑,/,*,+,–)
FZ(*)	= matriz con las funciones reconocidas (ABS,INT,COS,SIN,SQR,TAN)
CO(*)	= matriz teclas comando reconocidas
F$(*)	= matriz de las fórmulas introducidas
BP$	= beep de señalización
DR$	= unidad seleccionada
D1$	= cadena para comunicaciones disco
X,Y	= coordenadas del cursor
A0	= valor de la tecla pulsada
XX,YY	= coordenadas de la celda actual
D$	= contenido de la celda actual
PF$	= flag para cálculo
C$	= cadena de entrada (subrutina 3500)
L0	= longitud prevista para C$
R,C	= línea y columna en el tablero
ER	= flag de error
A$	= resultado del cálculo
RI,RF,RC	= parámetros para comando "C" (subrutina 15000)
F0$	= flag para el tipo de datos (1 = alfanuméricos; 2 = numéricos)
CI,CF,RF	= parámetros para el comando "M" (subrutina 30000)
NR	= registro actual
NF$	= fichero datos
AA$	= cadena I/O con disco
LR	= longitud registro

El tablero electrónico/Versión Philips VG 8010


```
4    REM ------------------
5    REM TABLERO ELECTRONICO
6    REM ------------------
7    LOCATE 0,0,0: KEY OFF
8    CLEAR 8000:DEFINTB-Y
9    H1$="E.G.S."
10   DATA"^","/","*","-","+"
20   DATA"ABS","INT","COS","SIN","SQR","TAN"
30   FOR I=1 TO 5:READ OP$(I):NEXT
35   DATA 70,65,78,73,67
40   FOR I=1 TO 6:READ FZ$(I):NEXT
50   FOR I=1 TO 5:READ CO(I):NEXT
60   GOSUB500:GOTO5000
500  REM ---------------------
530  SCREEN0:WIDTH40:CLS:BP$="O6L32CDE"
540  BA$=STRING$(40,32):BU$=STRING$(40,"_"):DIMB$(15)
550  FOR I=1 TO 15:FOR J=1 TO 24:B$(I)=B$(I)+LEFT$(BA$,10):NEXT J,I
570  RETURN
1000 REM --- ENCIENDE CURSOR ---
1010 LOCATEX,Y,1:RETURN
1100 REM --- APAGA CURSOR ---
1110 LOCATE X,Y,0:RETURN
1200 REM --- ESPERA CARACTER ---
1210 A$=INKEY$:IFA$=""THEN1210
1220 AO=ASC(A$):GOSUB1100:RETURN
1300 REM --- enciende celda ---
1310 LOCATEXX-1,YY-1:PRINTDD$
1320 XX=DX*8+2:YY=DY*2+2
1330 GOSUB9500:IF PF$<>"0" AND PF$<>" " THEN GOSUB 17000
1340 LOCATE XX-1,YY-1:PRINT D$:X=XX-1:Y=YY-1:GOSUB 1000:DD$=D$
1350 RETURN
2130 PRINT F$(1)
2700 REM --- centra en pantalla ---
2736 XT=(42-LEN(TI$))/2:LOCATEXT,YT:PRINTTI$;:RETURN
2900 REM --- elimina espacios ---
2930 Z2$="":A=LEN(Z$):FOR I=1 TO A:Z1$=MID$(Z$,I,1):IF Z1$<>" " THEN Z2$=Z2$+Z1$
2950 NEXT:Z$=Z2$:RETURN
3000 REM --- borra linea YT ---
3010 LOCATEO,YT:PRINTSPACE$(39);:RETURN
3500 REM --- entrada campo ---
3530 IC=0:C$=B$(CC):IF LO=0 THEN RETURN
3535 IF CC=0 THEN LOCATE XC,YC:PRINT MID$(BU$,1,LO)
3537 LOCATE XC,YC:PRINT C$;
3540 X=XC:Y=YC
3545 REM ---------------------------
3550 GOSUB 1000:GOSUB 1200:GOSUB 1100
3560 IF AO=30 OR AO=31 OR AO=27 THEN RETURN
3580 IF AO<>13 THEN 3620
3582 IF IC=0 THEN RETURN
3590 C$=MID$(C$,1,IC):A$=MID$(BA$,1,LO-IC):C$=C$+A$:A$=MID$(BU$,1,LO-IC)
3605 LOCATEXC+IC,YC:PRINTA$:RETURN
3620 IF AO<>29 THEN 3650
3630 IF IC=0 THEN PLAYBP$:GOTO 3550
3640 IC=IC-1:X=X-1:GOTO 3550
3650 IF IC=LO THEN PLAYBP$:GOTO 3550
3655 IF AO=0 THEN 3550
3660 IF AO=28 THEN IC=IC+1:X=X+1:GOTO 3545
3670 CH$=CHR$(AO):PRINTCH$;:C$=MID$(C$,1,IC)+CH$+MID$(C$,IC+2,39):IC=IC+1:X=X+1:GOTO3545
5000 REM --- MAIN ---
5010 GOSUB 8000:DD$="":XX=1:YY=1
5020 DX=C-C1:DY=R-R1
5030 GOSUB 1300:GOSUB 1200
5040 IF AO<>28 THEN 5070
5050 IF DX=4 AND C1<22 THEN C=C+1:C1=C1+1:GOSUB 8100:GOTO 5020
5060 IF DX<4 THEN C=C+1:GOTO 5020
5070 IF AO<>29 THEN 5100
5080 IF DX=0 AND C1>1 THEN C=C-1:C1=C1-1:GOSUB 8100:GOTO 5020
5090 IF DX>0 THEN C=C-1:GOTO 5020
5100 IF AO<>31 THEN 5130
5110 IF DY=9 AND R1<6 THEN R=R+1:R1=R1+1:GOSUB 8100:GOTO 5020
5120 IF DY<9 THEN R=R+1:GOTO 5020
5130 IF AO<>30 THEN 5160
5140 IF DY=0 AND R1>1 THEN R=R-1:R1=R1-1:GOSUB 8100:GOTO 5020
5150 IF DY>0 THEN R=R-1:GOTO 5020
5160 GOSUB 16000:GOTO 5020
8000 REM --- inicializa pantalla ---
8010 R$="":FOR I=65 TO 79:R$=R$+CHR$(I):NEXT I
8015 V$="FANIC"
8020 R1=1:C1=1:R=1:C=1:FOR I=2 TO 20:LOCATE 0,I-1:PRINT" |";:NEXTI
8025 LOCATEO,23:PRINTH1$;TAB(26)"F A N I C";
8100 REM --- SCROLL ---
8110 C2=C1:LOCATEO,0:PRINTLEFT$(BA$,40);
8120 LOCATEO,0:FORI=5TO37STEP8
8130 LOCATEI-5,0:PRINT"┌──";:PRINTUSING"##";C2;:PRINT"──┐";:C2=C2+1:NEXTI
8140 R2=R1:FORI=2TO20STEP2
8150 LOCATEO,I-1:PRINTMID$(R$,R2,1):R2=R2+1
8160 NEXT I
8170 Y=1:FOR I=R1 TO R1+9
8180 X=1:FOR J=C1 TO C1+4
8190 N1=1+9*(J-1):A$=MID$(B$(I),N1,9)
8195 FO$=LEFT$(A$,1):PF$=MID$(A$,2,1)
8200 D$=RIGHT$(A$,7):LOCATE X,Y
8210 PRINTD$:X=X+8:NEXTJ
8220 Y=Y+2:NEXTI:RETURN
9000 REM -------
9004 REM CALCULO
9008 REM -------
9016 L2=LEN(A$):C$="":FOR I=1 TO L2:CH$=MID$(A$,I,1)
9020 IFCH$<>" "THENC$=C$+CH$
9024 NEXTI:A$=C$:C$=""
9028 A$="+"+A$:GOTO9140
9032 REM --- CAL 2 ---
9036 ER=0:L=LEN(CA$):IFL<3THENRETURN
9040 FOR I=2 TO L-1:CH$=MID$(CA$,I,1):FOR JJ=1 TO 5:IF OP$(JJ)=CH$ THEN 9048
9044 NEXTJJ,I:ER=1:RETURN
9048 CH$=MID$(CA$,I+1,1):IF CH$="*" OR CH$="^" OR CH$="/" THEN ER=2:RETURN
9052 Z1=VAL(MID$(CA$,1,I-1)):Z2=VAL(MID$(CA$,I+1,40)):ONJJGOSUB9060,9064,9072,9076,9080:CA$=STR$
(Z1):IFZ1>=0THENCA$="+"+MID$(CA$,2,40)
9056 RETURN
9060 Z1=Z1^Z2:RETURN
9064 IF Z2=0 THEN ER=3:RETURN
9068 Z1=Z1/Z2:RETURN
9072 Z1=Z1*Z2:RETURN
9076 Z1=Z1-Z2:RETURN
9080 Z1=Z1+Z2:RETURN
9084 REM --- FORM 2 ---
9088 ER=0
9092 FOROP=1TO5:LO=LEN(FO$):KK=2:IFLO=1THENKK=1
9096 PO=0:P1=0:P2=0:FORIO=KKTOLO-1:CH$=MID$(FO$,IO,1):CH=ASC(CH$):IFCH$=OP$(OP)ANDPO=0THENPO=IO:
GOTO9112
9100 O=0:IF(CH=94ORCH<48)ANDCH<>46THENO=1
9104 IFO=1ANDPO=0THENP1=IO
9108 IFO=1ANDPO>0ANDIO-PO>1THENP2=IO:GOTO9120
9112 NEXTIO:IFPO>0THEN9120
9116 NEXTOP:RETURN
9120 IFP1=0THENP1=1:GOTO9128
9124 CH$=MID$(FO$,P1,1):IFCH$<>"+"ANDCH$<>"-"THENP1=P1+1
9128 P2=P2-1:IFP2<0THENP2=LO
9132 CA$=MID$(FO$,P1,P2-P1+1):GOSUB9036:IFER<>0THENRETURN
9136 GOSUB9256:FO$=MID$(FO$,1,P1-1)+CA$+MID$(FO$,P2+1,40):GOTO9088
9140 REM --- PAR 1 ---
9144 ER=0:LA=LEN(A$):IFLA<3THEN9164
9148 PI=0:PF=0:FORIA=1TOLA:CH$=MID$(A$,IA,1):IFCH$="("THENPI=IA
9152 IFCH$=")"THEN9180
9156 NEXTIA:IFPI>0THENER=4:RETURN
9160 FO$=A$:GOSUB9084:A$=FO$
9164 IFVAL(A$)=0THENA$="0"
9168 IFMID$(A$,2,1)="."THENA$=LEFT$(A$,1)+"0"+MID$(A$,2,80)
9172 IFLEFT$(A$,1)="+"THENA$=MID$(A$,2,80)
9176 RETURN
9180 IFPI=0THENER=4:RETURN
9184 PF=IA:FO$=MID$(A$,PI+1,PF-PI-1):GOSUB9084:IFER<>0THENRETURN
```

```
9188 IFPI>3THENGOSUB9208
9192 IFPI<=1THEN9200
9196 IFA<0ANDMID$(A$,PI-1,1)="-"THENPI=PI-1:FO$="+"+MID$(FO$,2,80)
9200 IFER<>0THENRETURN
9204 A$=MID$(A$,1,PI-1)+FO$+MID$(A$,PF+1,80):GOTO9144
9208 REM --- FUN 1 ---
9212 ER=0:CH$=MID$(A$,PI-1,1):IFCH$<"A"ORCH$>"Z"THENRETURN
9216 FU$=MID$(A$,PI-3,3):FORI=1TO6:IFFU$=FZ$(I)THEN9224
9220 NEXTI:ER=5:RETURN
9224 ZA=VAL(FO$):ONIGOSUB9228,9232,9236,9240,9244,9252:FO$=STR$(ZA):CA$=FO$:GOSUB9256:FO$=CA$:IF
ZA>=0THENFO$=MID$(FO$,2,40):PI=PI-3:RETURNELSEPI=PI-3:RETURN
9228 ZA=ABS(ZA):RETURN
9232 ZA=INT(ZA):RETURN
9236 ZA=COS(ZA):RETURN
9240 ZA=SIN(ZA):RETURN
9244 IFZA<0THENER=6:RETURN
9248 ZA=SQR(ZA):RETURN
9252 ZA=TAN(ZA):RETURN
9256 REM --- ESP 1 ---
9260 ZQ$=STRING$(64,"0"):L=LEN(CA$)
9264 FORIC=1TOL:IFMID$(CA$,IC,1)="E"THEN9272
9268 NEXT:RETURN
9272 N$=MID$(CA$,2,IC-2):L=LEN(N$)
9276 FORI=1TOL:IFMID$(N$,I,1)="."THENP=I:GOTO9284
9280 NEXT:P=L
9284 E=VAL(MID$(CA$,IC+1,3)):IFE<0THEN9316
9288 DD=L-P:IFDD=0THEN9308
9292 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80)
9296 IFDD<=ETHEN9308
9300 N$=MID$(N$,1,P+E-1)+"."+MID$(N$,P+E,80)
9304 DD=E
9308 N$=N$+MID$(ZQ$,1,E-DD)
9312 GOTO9332
9316 IFP=LTHENP=P+2-L:GOTO9324
9320 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80):L=1
9324 N$=MID$(ZQ$,1,-E-L+1)+N$
9328 N$=MID$(N$,1,P-1)+"."+MID$(N$,P,80)
9332 CA$=LEFT$(CA$,1)+N$
9336 RETURN
9400 REM --- VAR 1 ---
9405 Z$=A$:GOSUB2900:A$=Z$
9410 LA=LEN(A$):IFLEFT$(A$,3)="SUM"THENGOSUB14000:RETURN
9415 FORI=1TOLA
9420 CH$=MID$(A$,I,1):IFCH$>="A"ANDCH$<="Z"THEN9440
9430 NEXTI:GOTO9000
9440 PI=I:C$=MID$(A$,I+2,1):IFC$>="A"ANDC$<="Z" THEN I=I+2:GOTO 9430
9450 IFCH$>"O"THENER=7:RETURN
9460 R=ASC(CH$)-64:C=VAL(MID$(A$,PI+1,1)):IFC=0THENER=7:RETURN
9470 GOSUB9500:Z$=D$:GOSUB2900
9480 A$=MID$(A$,1,PI-1)+Z$+MID$(A$,PI+2,80)
9490 GOTO9410
9500 REM --- LECTURA CELDA ---
9510 N1=1+9*(C-1):B$=MID$(B$(R),N1,9)
9520 FO$=LEFT$(B$,1):PF$=MID$(B$,2,1)
9530 D$=RIGHT$(B$,7):RETURN
9600 REM --- ESCRITURA CELDA ---
9610 N1=9*(C-1):N2=N1+10
9620 A1$=MID$(B$(R),1,N1):A2$=MID$(B$(R),N2,235)
9630 D$=LEFT$(D$,7)
9640 IFFO$="1"THEND$=D$+MID$(BA$,1,7-LEN(D$))
9650 IFFO$="2"THEND$=MID$(BA$,1,7-LEN(D$))+D$
9660 B$(R)=A1$+FO$+PF$+D$+A2$:RETURN
14000 REM --- SUM ---
14010 ER=0:FORI=5TOLA-1
14020 IFMID$(A$,I,1)="."THEN14040
14030 NEXT:ER=10:A$="ERR=10":RETURN
14040 A1$=MID$(A$,5,I-5):A2$=MID$(A$,I+1,LA-I-1)
14060 RI$=LEFT$(A1$,1):RF$=LEFT$(A2$,1)
14080 CI=VAL(MID$(A1$,2,2)):CF=VAL(MID$(A2$,2,2))
14090 RI=0:RF=0:A1=0
14100 FOR I=1 TO 15:CH$=MID$(R$,I,1)
14110 IF CH$=RI$ THEN RI=I
14120 IF CH$=RF$ THEN RF=I
14130 NEXT I
14140 IF RI=0 OR RF=0 OR CI=0 OR CF=0 THEN ER=10:A$="ERR=10":RETURN
14145 IF RI<>RFANDCI<>CFTHENRI=0:GOTO14140
14150 IF CI=CF THEN 14190
14160 R=RI:FOR I=CI TO CF:C=I
14170 GOSUB 9500:A1=A1+VAL(D$)
14180 NEXT I:GOTO 14220
14190 C=CI:FOR I=RI TO RF:R=I
14200 GOSUB 9500:A1=A1+VAL(D$)
14210 NEXT I
14220 A$=STR$(A1):RETURN
15000 REM --- COPIA ---
15010 LOCATE0,23:PRINT"(ESC para salir)";
15020 LOCATE0,21:LO=2
15030 PRINT"LINEA INICIAL ?":XC=15:YC=21
15040 B$(0)="":CC=0:GOSUB 3500:IF A0=27 THEN 15150
15050 RI=VAL(C$):YT=21:GOSUB3000
15060 LOCATE0,21:PRINT"LINEA FINAL ?"
15070 XC=13:GOSUB3500:IFA0=27THEN15150
15080 RF=VAL(C$):LOCATE0,21
15090 PRINT"LINEA PRINCIPIO COPIA?":XC=23
15100 GOSUB 3500:IF A0=27 THEN 15150
15110 RC=VAL(C$):GOSUB3000
15120 IFRI<=0ORRF<=0ORRC<=0ORRI>15ORRF>15ORRC>15ORRI>RFTHEN15020
15130 FORI=RITORF:B$(RC)=B$(I)
15140 RC=RC+1:NEXTI:GOSUB8000
15150 YT=21:GOSUB3000:YT=23:GOSUB3000:RETURN
16000 REM --- COMANDOS ---
16010 FOR I=1 TO 9
16020 IF CO(I)=A0 THEN 16035
16030 NEXT I:RETURN
16035 GOSUB 16500
16040 ONIGOSUB17500,18000,18500,20000,15000
16050 LOCATE0,23:PRINTH1$;TAB(26)"F A N I C";
16060 RETURN
16500 REM --- PRESENTA COMANDO ---
16510 LOCATE 26,23
16520 REM
16530 PRINTSTRING$(I*2-2," ")+MID$(V$,I,1)+STRING$(10-I*2," ");:LOCATE0,0:RETURN
17000 REM --- CALCULO ---
17010 PF=VAL(PF$):IFPF=0ORFO$<>"2"THENRETURN
17030 A$=F$(PF):RR=R:CC=C:GOSUB9400
17035 L=LEN(A$):IFER<>0THENA$="ERR="+STR$(ER):GOTO17110
17040 IFL<8THEN17110
17050 IFVAL(A$)=INT(VAL(A$))THENA$="EEEEEEE":GOTO17110
17060 FOR I=L TO 1 STEP-1
17070 CH$=MID$(A$,I,1):IFCH$="."THEN17090
17080 NEXTI:ER=10:RETURN
17090 IFI>8THENA$="EEEEEEE":GOTO17110
17100 A$=LEFT$(A$,7)
17110 R=RR:C=CC:GOSUB9500:D$=A$
17120 GOSUB 9600:RETURN
17500 REM --- INTRODUCCION FORMULA ---
17510 FORI=1TO8:IFF$(I)=""THEN17530
17520 NEXT
17530 PF=VAL(PF$):IFPF=0THENPF=I:PF$=RIGHT$(STR$(I),1)
17535 FO$="2"
17537 B$(0)=F$(PF):CC=0
17540 LOCATE0,21:PRINT"FORM.:"
17550 LO=30:XC=7:YC=21:CC=0:GOSUB3500:Z$=C$:GOSUB2900
17555 IFZ$=""THENPF$="0"
17556 GOSUB9600
17560 F$(PF)=Z$:YT=21:GOSUB3000:B$(0)="":GOSUB17000:RETURN
18000 REM--- INTRODUCCION ALFANUMERICOS
18005 PF=VAL(PF$):F$(PF)=""
18010 PF$="0":FO$="1":CC=0:Z$=D$:GOSUB2900:B$(0)=Z$
18020 LOCATE0,21:PRINT"DATO: "
18030 LO=7:XC=6:YC=21:GOSUB3500
18040 D$=C$:GOSUB9600
18050 YT=21:GOSUB3000:RETURN
18500 REM ---- INTRODUCCION NUMERICOS
18510 FO$="2":LOCATE0,21:PRINT"DATO: "
18520 CC=0:Z$=D$:GOSUB2900:B$(0)=Z$:LO=7:XC=6:YC=21:GOSUB3500
18525 Z$=C$:GOSUB2900:C$=Z$
18530 FOR I=1 TO LEN(C$):CH$=MID$(C$,I,1)
18535 IF(CH$<"0"ORCH$>"9")ANDCH$<>"."ANDCH$<>"-"THEN PLAYBP$:GOTO 18520
18540 NEXT I:D$=C$:GOSUB 9600:GOSUB 1300
18550 YT=21:GOSUB 3000:RETURN
19000 REM --- INSERCION LINEAS ---
19010 FORI=15TOR+1STEP-1
19020 B$(I)=B$(I-1):NEXTI
19030 B$(R)="":FOR I=1 TO 6:B$(R)=B$(R)+BA$
19040 NEXT I:GOSUB 8100:RETURN
20000 REM --- INSERCION LINEA Y COLUMNA
20010 LOCATE 0.21:PRINT"L C E";
```

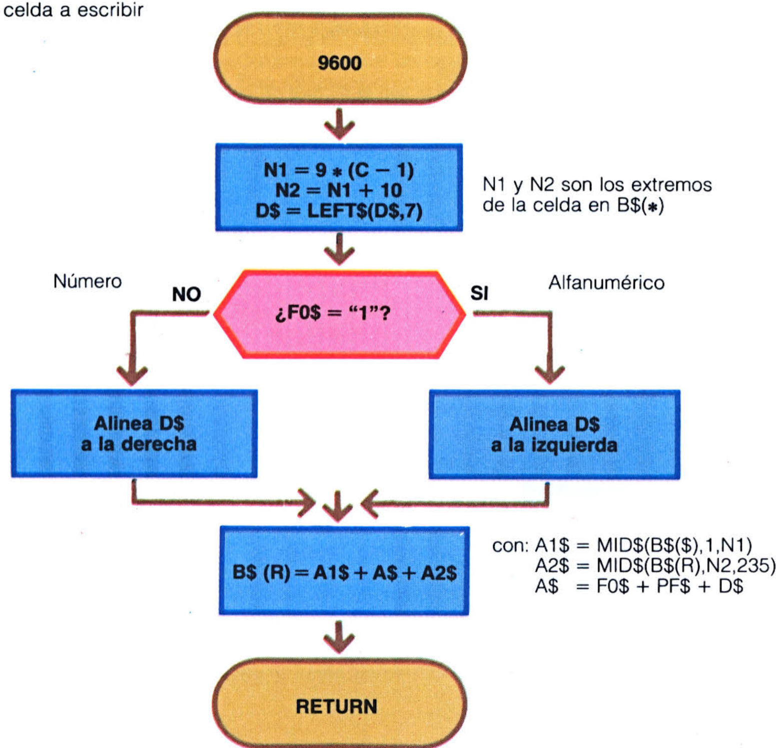
```
20020 GOSUB 1200:IF A0<> 76 AND A0<>67 AND A0<>69 THEN 20020
20030 IF A0=76 THEN LOCATE 0,21:PRINT"* L *";
20040 IF A0=67 THEN LOCATE 0,21:PRINT"* C *";
20050 IF A0=69 THEN 20080
20060 IF A0=76 THEN GOSUB 19000:GOTO 20080
20070 GOSUB 20500
20080 YT=21:GOSUB 3000:RETURN
20500 REM --- INSERCION COLUMNAS ---
20510 K=C:FOR IC=25 TO K STEP-1
20520 C=IC:GOSUB 9500:C=IC+1
20530 GOSUB 9600:NEXT IC
20540 PF$="0":F0$="1":D$=SPACE$(7):C=K
20550 GOSUB 9600:GOSUB 8100:RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - PHILIPS VG 8010

B$(*)	= matriz de las cadenas para el tablero
OP$(*)	= matriz con los símbolos de los operadores (↑,/,*,+,–)
FZ(*)	= matriz con las funciones reconocidas (ABS,INT,COS,SIN,SQR,TAN)
CO(*)	= matriz teclas comando reconocidas
F$(*)	= matriz de las fórmulas introducidas
BP$	= beep de señalización
X,Y	= coordenadas del cursor
A0	= valor de la tecla pulsada
XX,YY	= coordenadas de la celda actual
D$	= contenido celda actual
PF$	= flag para cálculo
C$	= cadena de entrada (subrutina 3500)
L0	= longitud prevista para C$
R,C	= línea y columna en el tablero
ER	= flag de error
A$	= resultado del cálculo
RI,RF,RC	= parámetros para comando "C" (subrutina 15000)
F0$	= flag para el tipo de datos (1 = alfanuméricos; 2 = numéricos)

ESCRITURA DE UNA CELDA

Entradas:
L,C = Coordenadas en la celda a escribir
D$, F0$, PF$



Gestión de los comandos. La subrutina de gestión de los comandos debe adquirir el carácter que representa la función deseada, comprobar su existencia entre las previstas (matriz CO(*)) y activar la correspondiente subrutina de desarrollo.
En el diagrama de flujo, estas subrutinas no se han representado porque su número y su variedad dependen de la máquina utilizada o, mejor dicho, de la estructura y las posibilidades del Basic y de la cantidad de memoria disponible.
El método para insertar un comando es el siguiente:

1 / Primero hay que incluir la letra (código) de identificación del comando en la matriz CO(*)
2 / Preparada la rutina para su desarrollo, debe completarse la instrucción ON I... con su dirección (número de línea).

En este punto, el comando se activa y puede ser utilizado. Sin embargo, hay que tener en cuenta que, en algunos casos, la inclusión de una nueva función tiene como consecuencia la necesidad de importantes modificaciones en las otras subrutinas. Por ejemplo, una función muy frecuente en los tableros electrónicos da la posibilidad de variar la longitud de los datos presentados (o sea D$). Activando esta implantación hay que modificar los valores numéricos utilizados, como en la 8100 (desplazamiento por la pantalla) y en todas las subrutinas que utilizan D$.
Algunas funciones están incluidas en los programas y se describen más adelante. Los correspondientes diagramas se han omitido, por lo que hay que consultar los listados.

Introducción (subrutina 3500). En el main (pág. 2420) se lee un carácter, en base al cual se activan 3 subrutinas. Las primeras dos para el desplazamiento del cursor o la gestión de los comandos, y la última para la introducción de los datos. A su vez, esta subrutina prevé 3 posibilidades, siempre en función del carácter (CH$) introducido en el main:

1 / El usuario desea escribir una fórmula. Este caso se indica con un valor particular de CH$ (por ejemplo, el símbolo F)
2 / El dato está constituido por una cadena alfanumérica. Este caso también se indica con un valor particular de CH$ (por ejemplo, el símbolo A)

PRESENTACION DEL CONTENIDO DE UNA CELDA

En llamada:

H2 = Flag de desplazamiento (0 = N.D., 1 = I, 2 = D)
L = Línea
C = Columna
H3 = Flag que activa la lectura de la celda

1300

Lectura celda 9500 — Esta subrutina sólo se llama si H3 = 1 (ver subrutina 3000)

H2 = 1 — Salva el flag de presentación

¿H2 = 0? — SI → **¿F0$ = "1"?** — SI → **H2 = 1** — El campo es alfanumérico, va a la izquierda

¿F0$ = "1"? — NO → **H2 = 2** — Numérico, va a la derecha

¿H2 = 0? — NO →

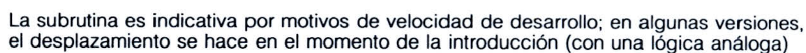
¿H2 = 1? — SI → **Va a la izquierda**

¿H2 = 1? — NO → **Va a la derecha**

D$

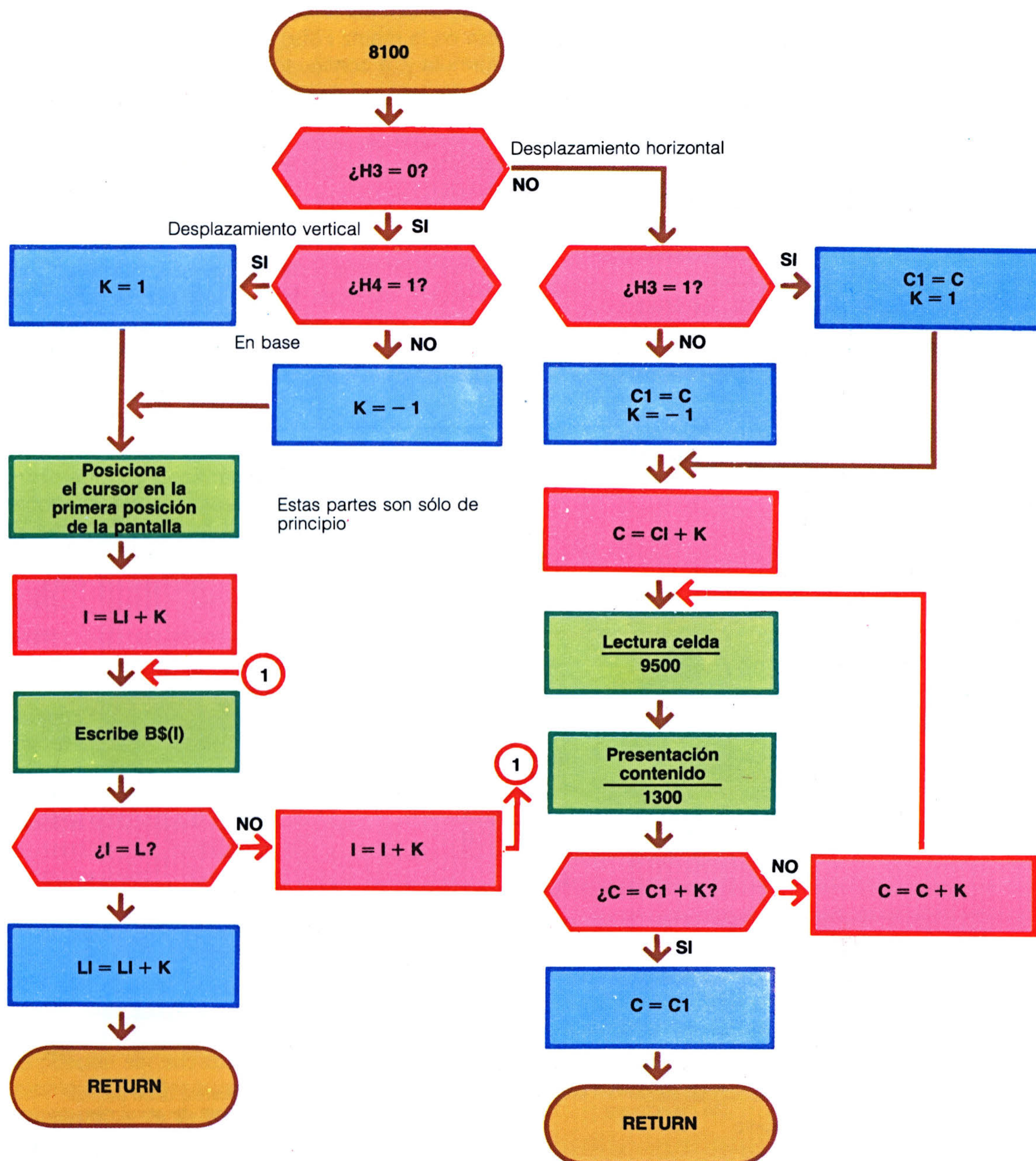
H3 = 1 — Predispone el flag de lectura celda

RETURN


La subrutina es indicativa por motivos de velocidad de desarrollo; en algunas versiones, el desplazamiento se hace en el momento de la introducción (con una lógica análoga)

TRASLACION DE LA PANTALLA

En llamada:
CI = Primera columna vídeo
LI = Primera línea vídeo
L = Línea de final de traslación
C = Columna de final de traslación
H3 = Flag de traslaciones horizontales
H4 = Flag de traslaciones verticales



La subrutina de escritura de B$(I) no se ha representado porque su desarrollo depende de la máquina utilizada. Sin embargo, puede estar constituida por: lectura y presentación de la cadena D$, o sea por las rutinas 9500 y 1300

N.B.: El diagrama sólo es indicativo. Hay que consultar los listados

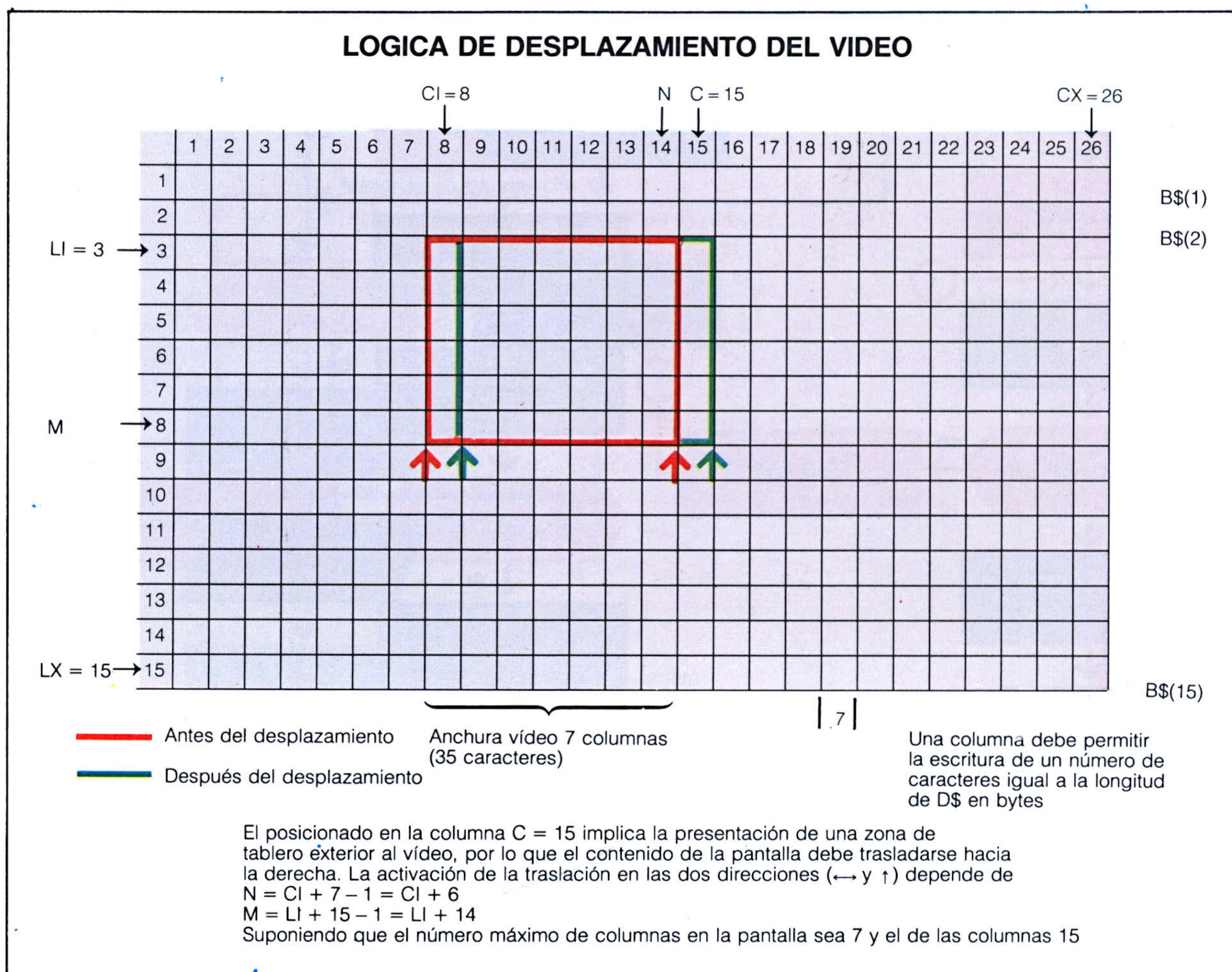
3 / Si no se cumple ninguna de las condiciones anteriores, el programa supone que el dato está constituido por valores numéricos.

En el diagrama, cada uno de los 3 posibles casos se trata con una subrutina, aunque la diferencia entre valor alfanumérico y numérico puede obtenerse simplemente activando un flag que sirva de control. Efectivamente, los caracteres introducidos son transferidos en los dos casos a la variable D$ y, de ésta, al buffer B$(R) en la columna C (L y C son las coordenadas de la celda actual).
La única diferencia entre ambos casos es el control sobre los valores introducidos para evitar que el usuario, por error, escriba un carácter alfabético en un campo numérico. Este control puede subordinarse fácilmente al valor de un flag.
En el diagrama y en los listados, se ha preferido separar ambos casos, incluso si esto produce la duplicación de algunas instrucciones, para obtener una mayor estructuración y una mejor elasticidad del programa.

Introducción fórmulas. La subrutina es llamada por la 16500 introduciendo el carácter de reconocimiento de fórmula (por ejemplo, el símbolo F) y realiza las funciones:

- Lectura de un carácter
- Si es diferente de la tecla RETURN, lo acumula en la cadena A$ que, al final de la introducción, contendrá la fórmula
- Terminada la introducción, el usuario pulsa la tecla RETURN y la subrutina pasa a la fase de elaboración: inicialmente se busca la primera posición vacía en la matriz F$(*) en la que hay memorizada la fórmula y, a continuación, se desarrolla el cálculo y se presenta el resultado.

En el diagrama no se ha representado la función de presentación del carácter, pero sí está en los listados y consiste en la lectura, carácter por carácter, de todo lo introducido. Además, para simplificar el diagrama, no se ha incluido en él ningún control sobre los datos introducidos (el control parcial se realiza en la subrutina de cálculo), mientras que es una buena norma comprobar los datos a medida que se van introduciendo, por lo menos formalmente. En esta aplicación, los valores que pueden ser aceptados son:

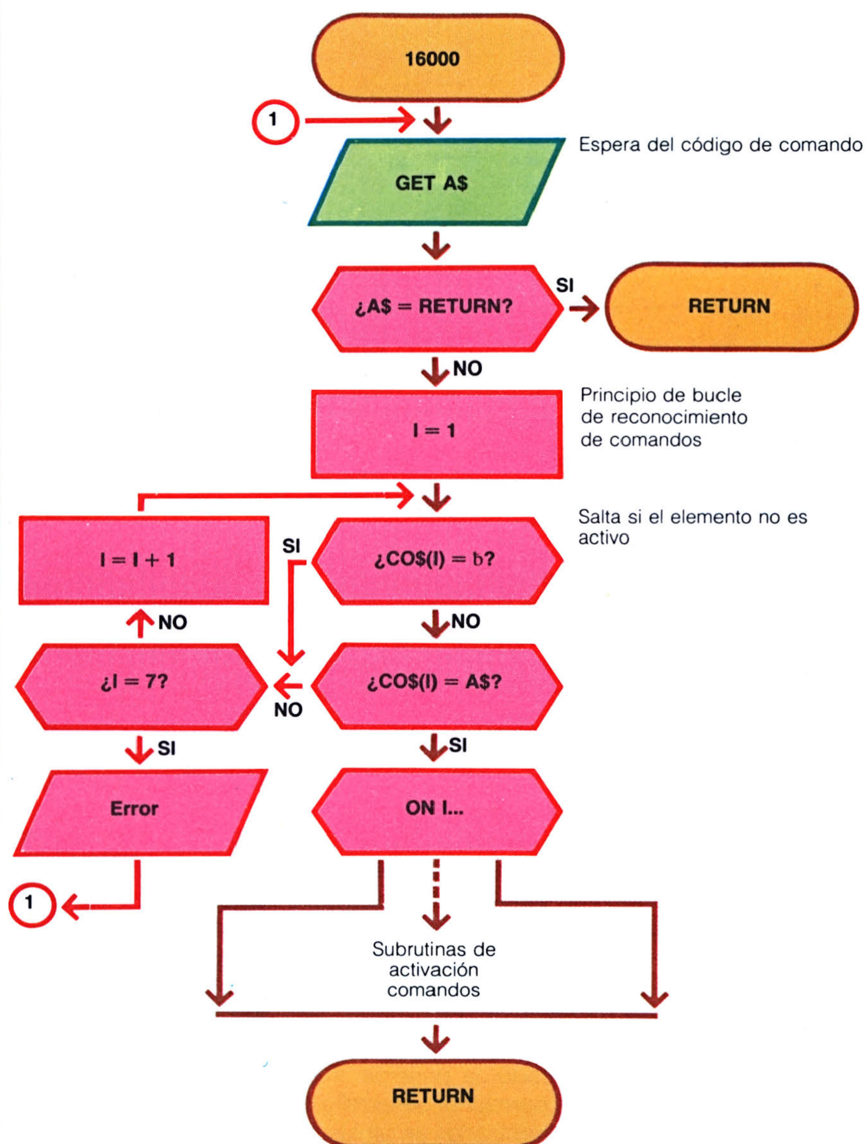


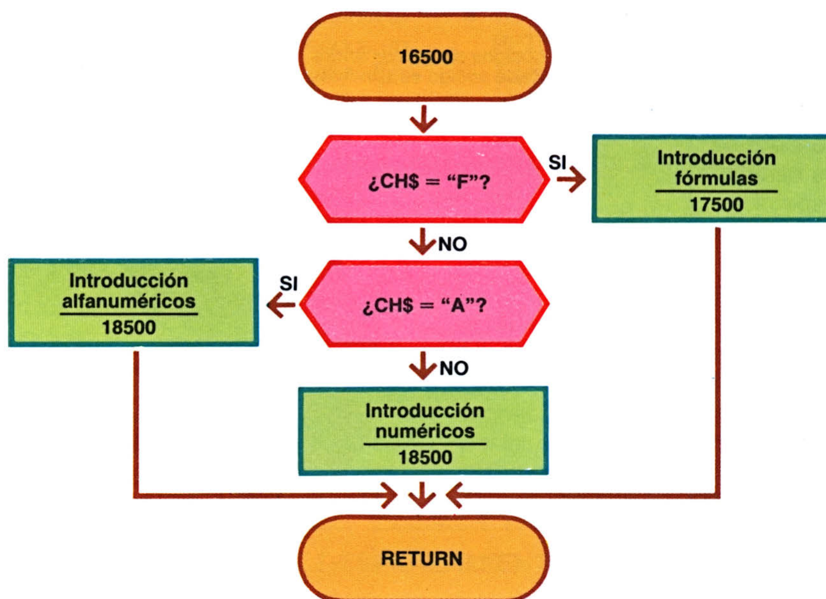
El posicionado en la columna C = 15 implica la presentación de una zona de tablero exterior al vídeo, por lo que el contenido de la pantalla debe trasladarse hacia la derecha. La activación de la traslación en las dos direcciones (⟷ y ↑) depende de
N = CI + 7 − 1 = CI + 6
M = LI + 15 − 1 = LI + 14
Suponiendo que el número máximo de columnas en la pantalla sea 7 y el de las columnas 15

1 / Numéricos, hasta un máximo de 5 cifras
2 / Alfanuméricos, sólo si constituyen el nombre de una función prevista
3 / Los siguientes símbolos: ↑ (elevación a potencia), / (división), ∗, +, – y los paréntesis "(" ")".

El control, excepto del segundo punto, es muy sencillo: se trata de descartar los caracteres que ni son números ni están comprendidos entre los siete símbolos ↑, ∗, /, +, –, (,).
Sin embargo, la necesidad de tener que reconocer los nombres de las funciones (constituidos por 3 letras) genera una complicación de los controles, porque al presentarse un carácter alfanumérico, antes de aceptarlo o descartarlo hay que esperar que el eventual nombre de la función se haya completado. Es decir, para reconocer si se ha introducido una función, hay que leer los datos en grupos de a 3, comprobando la eventual correspondencia con un elemento de la matriz. Además es necesario determinar cuándo el carácter forma parte de una dirección; por ejemplo, si se quiere multiplicar el contenido de la celda A1 (columna A, línea 1) por el valor numérico 5, la fórmula introducida será A1 ∗ 5: la subrutina debe poder reconocer el direccionamiento. Téngase en cuenta que este reconocimiento sólo es con el fin del diagnóstico de introducción, porque la fórmula debe escribirse siempre con las direcciones; los valores correspondientes sólo pueden ser sustituidos en el momento del desarrollo.
Resumiendo, para tener un diagnóstico completo en la fase de introducción, hay que realizar estos controles:

1 / El valor introducido es un número o un símbolo: se acepta





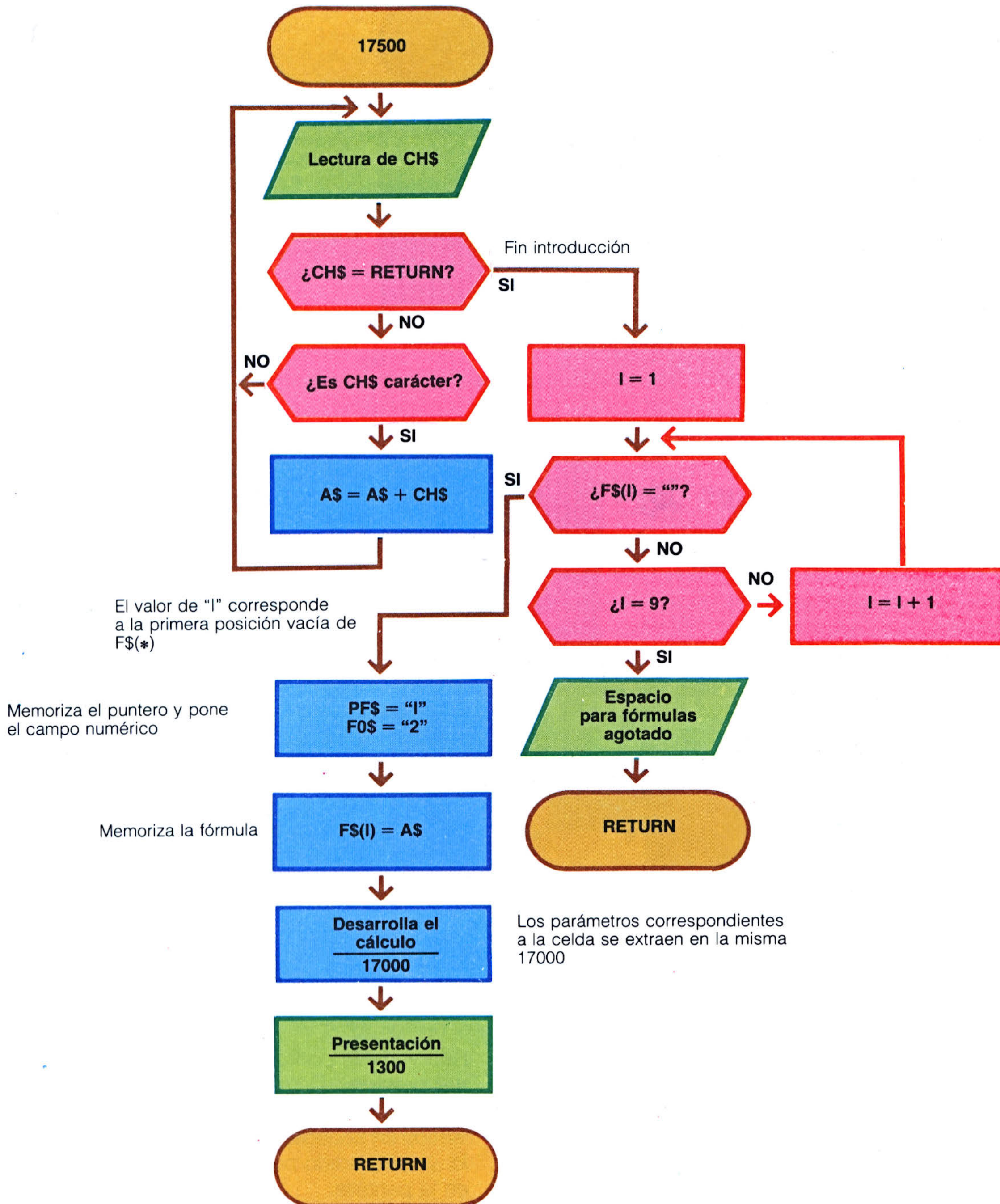
2 / El valor es una letra: se tienen dos casos:

- La letra va seguida de un número, se trata de una dirección
- La letra va seguida de otra letra, hay que esperar el 3er carácter del grupo para determinar si las 3 letras resultantes constituyen un nombre de función.

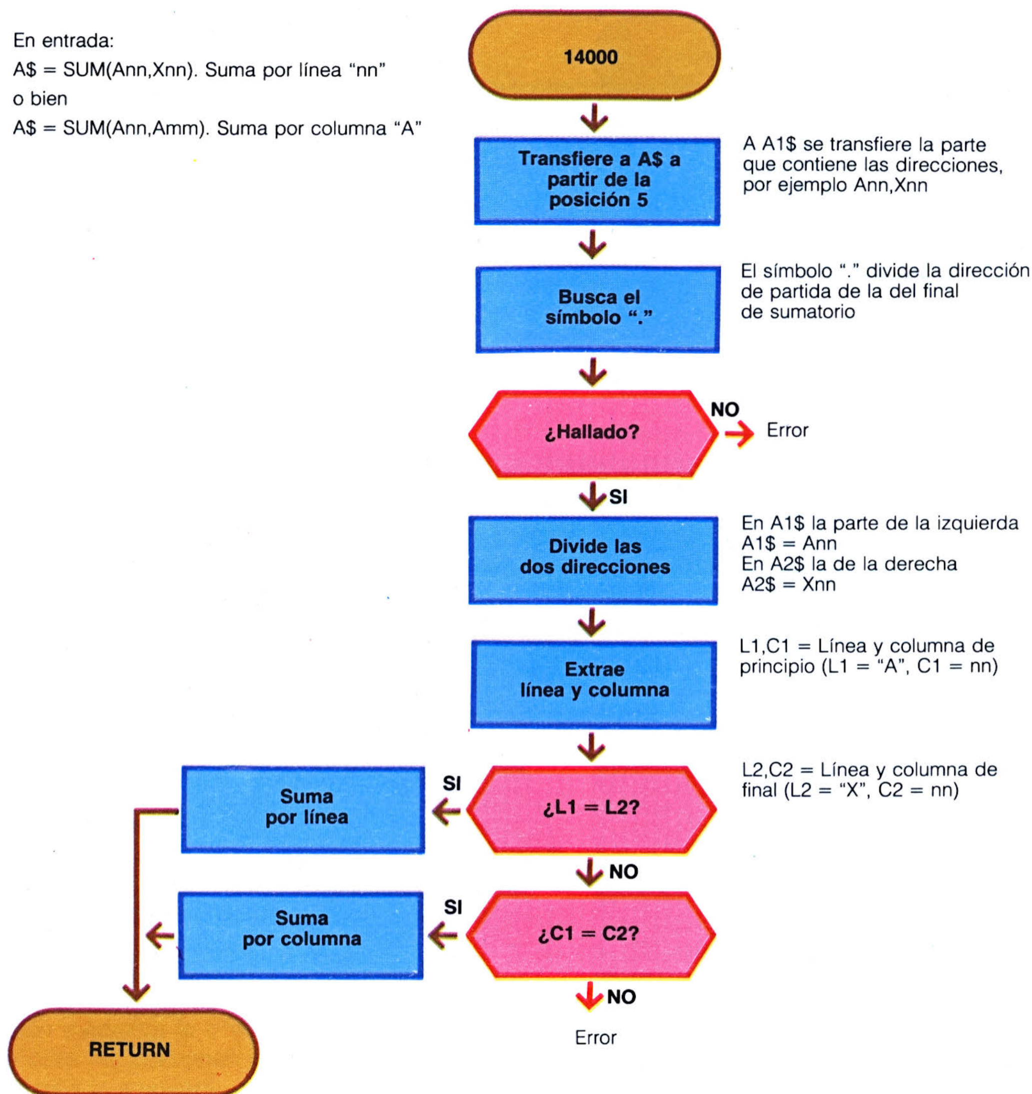
Obsérvese que la lógica de identificación de una dirección mediante la combinación letra + número (la letra para las columnas y el número para las líneas) está limitada a un tablero con las columnas marcadas con una sola letra, mientras que en el caso más general puede ser necesario un número mayor y, por tanto, con identificación de 2 letras, con ulteriores complicaciones en las funciones de control.

Funciones. Como se ha indicado, la principal finalidad del tablero electrónico es permitir el desarrollo de cálculos, incluso complejos, organizados en tablas. La subrutina de cálculo presentada permite el desarrollo de cálculos aritméticos, incluidas algunas funciones, que pueden ser tanto las del Basic como particulares, no previstas en el lenguaje «huésped». Las funciones previstas en el tablero electrónico deben declararse todas incluyendo, con una DATA, su nombre en la matriz adecuada, y se activan mediante subrutinas escritas en Basic, por lo que es muy sencillo idear nuevas funciones escribiendo el desarrollo en una subrutina dedicada.
Por ejemplo, suponiendo que se quiere incluir la función SQR(N) en el tablero (SQR es el nombre simbólico de la extracción de la raíz cuadrada del número N), basta con escribir su sigla (SQR) en la matriz de los nombres; la subrutina de desarrollo analiza la cadena que contiene el cálculo a desarrollar, extrae las letras SQR y, reconocido este nombre entre los previstos, activa una subrutina de desarrollo a la que se refiere la única instrucción SQR(N) que realiza la función pedida. De manera análoga pueden crearse funciones no previstas en el Basic: por ejemplo, para efectuar un sumatorio de varios valores en los tableros electrónicos se utiliza el código SUM seguido de las direcciones de partida y de final de la suma. La lógica a seguir es la misma, con la única diferencia del desarrollo de las funciones, que en este caso está al cuidado de la subrutina porque no está incluido entre los previstos en el Basic.
A título de ejemplo, en la página siguiente se ha representado el diagrama de la subrutina que activa las funciones, mientras que en la página 2444 hay el de la subrutina que realiza el sumatorio de los contenidos de las celdas especificadas en la instrucción SUM.
Los diagramas son a nivel funcional y los detalles pueden verse en los listados. La subrutina 17500, una vez reconocido un carácter no numérico, activa el bucle

INTRODUCCION DE FORMULAS



En el diagrama no se ha representado el bloque que ejecuta la presentación del carácter introducido



para la determinación del nombre de la función (este desarrollo es simplificado, puesto que en realidad hay que insertar un control para comprobar que el carácter no pertenezca a una dirección).
Como los nombres de las funciones son de tres letras, basta con leer los dos caracteres que siguen a CH$ en A$ para extraer la sigla a examinar. Determinada la posición de la sigla en la matriz de los nombres, puede activarse la subrutina de desarrollo.
A este propósito hay que observar que el valor resultante del cálculo debe escribirse siempre en la misma variable, sea cual fuere la subrutina que se esté empleando.

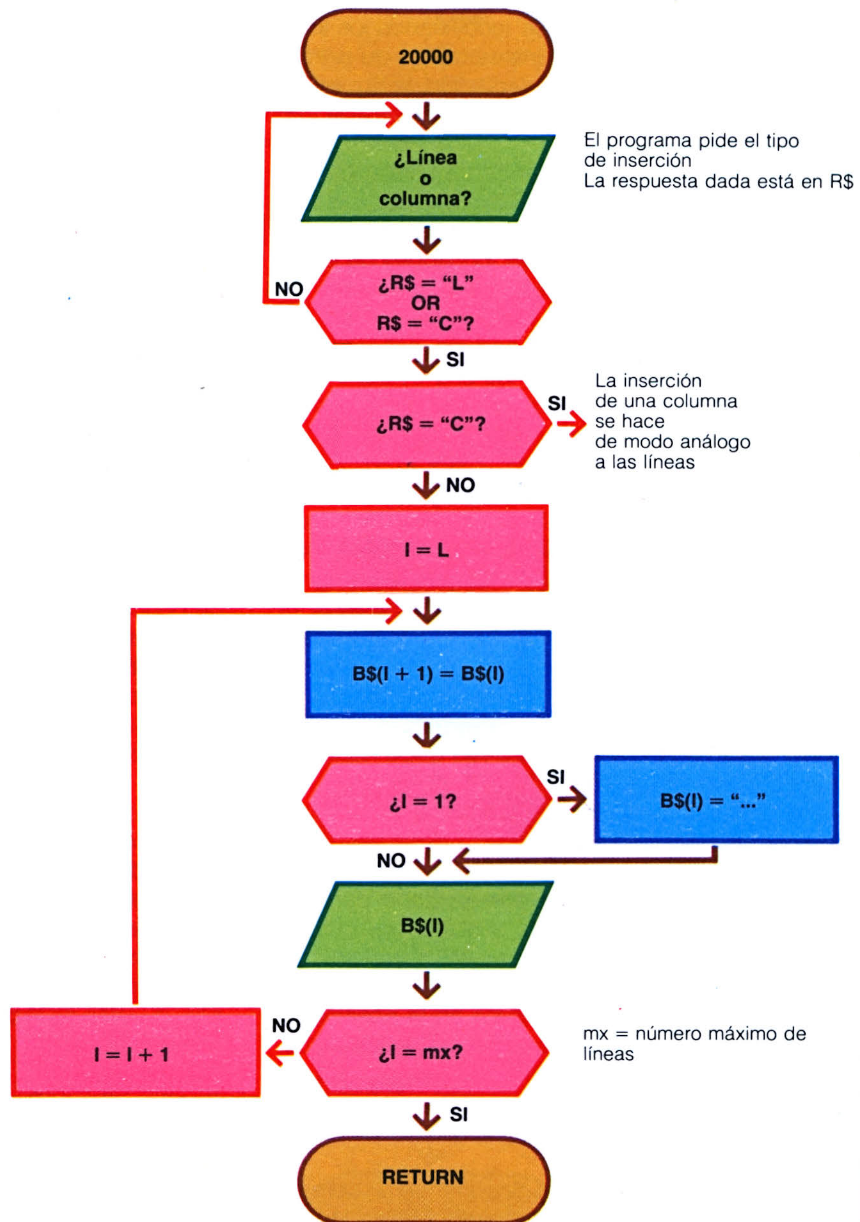
Esta variable podrá ser la misma en todas las subrutinas previstas. De esta manera, basta con transferir dicho valor a A$ (naturalmente, después de haberlo convertido en cadena) para obtener el resultado del cálculo, ya preparado para la transferencia en una celda o en la pantalla.
La figura de arriba muestra el desarrollo de la función SUM. Ante todo hay que definir la sintaxis según la cual debe escribirse la función y que será utilizada como lógica de desarrollo. En este ejemplo, la forma elegida es:

SUM (Dirección de partida.Dirección de final)

El símbolo "." es muy importante. Divide las dos direcciones y, de hecho, a la entrada de la subrutina, la primera función es precisamente la separación de las dos direcciones y se basa en la posición del punto. La parte de la izquierda se memoriza en A1$ y la de la derecha se memoriza en A2$ (obsérvese que A1$ también se utiliza para extraer todo el contenido de los paréntesis; por tanto, es una asignación provisional que puede perderse).
De estas dos cadenas (A1$ y A2$) se aíslan sucesivamente los valores de línea y de columna que identifican respectivamente las celdas de principio y de final.
Por tanto, sólo queda por activar un bucle que tome el contenido de cada celda comprendida entre las dos direcciones y los totalice en la variable Z.
Obsérvese que el sumatorio puede realizarse de dos maneras: por línea o por columna. En el primer caso, el bucle extrae las celdas incrementando el índice de línea (o sea las extrae de los distintos buffers B$(*)), mientras que en el segundo, es el índice de columna el que se extrae de las distintas posiciones del mismo buffer B$(*).

Comandos. La gestión completa del tablero electrónico requiere la presencia de algunas subrutinas para el desarrollo de funciones particulares, diferentes de las de desarrollo de cálculos. Estas funciones, que no deben confundirse con las matemáticas, pueden dividirse en algunos grupos principales:

- Funciones relativas al formato de presentación de los datos
- Comandos para la modificación de la estructura del tablero
- Gestión del disco
- Comandos particulares.



A continuación se analizan algunas funcionalidades de cada grupo, si bien en los listados no están todas. Por tanto, el análisis que sigue, en algunos casos, constituye la guía para la interpretación de los listados y, en otros, es una pista con la que pueden desarrollarse las implantaciones.

Formatos de presentación. La estructura representada presupone una longitud máxima del contenido de cada celda igual a 7 bytes, pero nada impide ampliar este campo en función de las propias necesidades específicas.
La ampliación puede obtenerse modificando los listados y precisamente todas las instrucciones de extracción, de memorización y de presentación de la cadena D$ (siempre teniendo en cuenta la limitación de la longitud total de cada línea).
La estructura de tablero presentada utiliza un cierto número de buffers para memorizar las distintas celdas. Cada buffer corresponde a una línea, por lo que su longitud es igual a la suma de las longitudes de todas las celdas presentes en la línea, o sea, a la suma de las amplitudes, en bytes, de las columnas.
El buffer de línea (B$(*)) es una cadena, por lo que su longitud total no puede superar los 255 bytes.
Con la estructura de celda vista (total de bytes, 2 para los flags y 7 para el valor), el número máximo de celdas y, por tanto de columnas, es: 234/9 = 26.
Al aumentar la longitud de cada celda se tiene una disminución proporcional del número de columnas. Además, la pantalla puede contener un número limitado de caracteres (generalmente, en este tipo de máquinas es de 40), por lo que utilizando una anchura excesiva de columna se tiene la presentación de un número de datos limitado.
Para obviar estos inconvenientes es útil parametrizar la forma de presentación, o sea el número de caracteres presentados en cada celda. Para ello debe haber un comando adecuado que, al activarlo, pregunte el número de caracteres a presentar y adecue las distintas subrutinas a la nueva situación.
Sin embargo, hay que tener en cuenta que esta implantación no es tan sencilla como puede parecer, porque también necesita una modificación de la subrutina de presentación de los ejes, del desplazamiento del cursor, del deslizamiento de la pantalla, etc.
Otra opción, muy utilizada, consiste en la posibilidad de modificar el formato de presentación de los datos, aunque respetando su longitud.
En los 7 bytes previstos para el valor de la celda puede memorizarse un número cualquiera, incluso con varios decimales.
En algunas aplicaciones es útil disponer de un comando que pueda limitar, y eventualmente anular, la presentación de los decimales.
El desarrollo de esta función es mucho más sencillo que el anterior, porque corresponde exclusivamente a la subrutina de presentación (1300) y permite extraer únicamente la parte que interesa de la cadena D$.
Operativamente, la activación de este comando requiere la inclusión del código en la matriz CO(*), la preparación de la subrutina de gestión y la modificación del bloque de presentación (D$) (incluido en la 1300).
Por ejemplo, suponiendo que se quiera asignar la letra F (formato) como código de activación de la función, el código deberá incluirse en CO(*) y, en la 1200, escribir la rutina de gestión que consiste en la activación de un flag que después se utilizará en la 1300 en la fase de presentación de D$.
Naturalmente, esta subrutina, además de activar el flag, debe pedir el número de cifras que se quieren presentar, valor que puede incluirse en el flag, por ejemplo inicializando al valor 7 (máxima longitud prevista) y modificándolo de acuerdo con la introducción.
En la 1300, la implantación consiste en extraer y presentar un número de cifras igual al valor del flag.

Modificaciones de la estructura del tablero. Con este tipo de comando se activan dos funciones:

1 / Inserción de una nueva línea
2 / Inserción de una nueva columna.

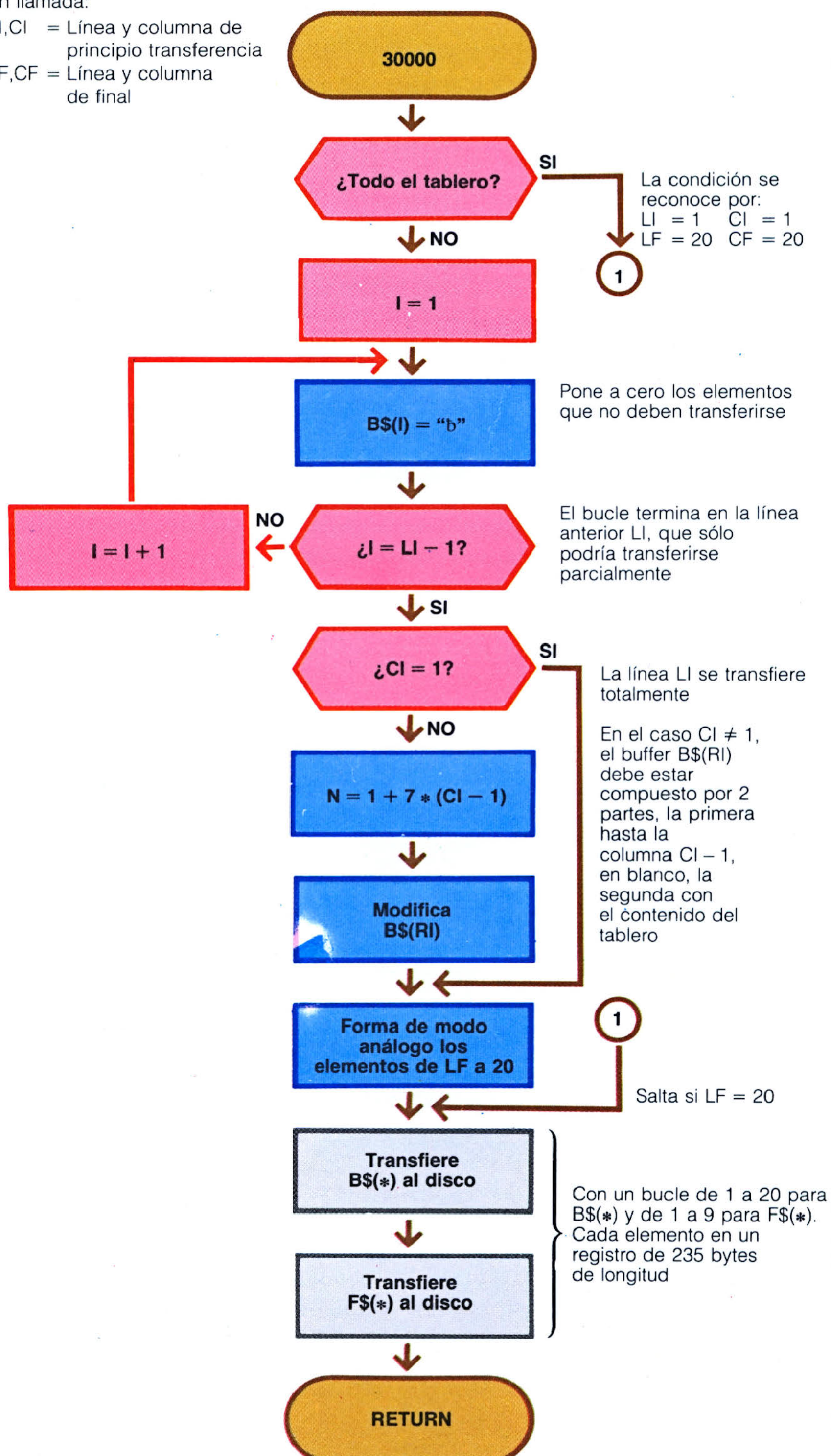
Naturalmente, es posible implantar el programa con muchas otras funciones, tales como: la subdivisión en ventanas, la protección de la escritura, etc., que, sin embargo, tienen un carácter secundario y, por tanto, no se tratarán en esta versión.
En la página anterior se ha representado el diagrama de principio de la subrutina de inserción de línea o de columna.
En sustancia se trata de transferir los contenidos de B$(*) de manera que quede una línea o una columna completamente en blanco. El diagrama está muy simplificado y puede utilizarse únicamente como indicación de una metodología.
En particular, y para abreviar, la parte correspondiente a la inserción de una columna no se muestra, pero puede desarrollarse conceptualmente de manera análoga a la utilizada para las líneas, aunque además requiere un bucle de traslación de los distintos campos de una misma línea.
Este bucle puede desarrollarse utilizando las subrutinas ya vistas de lectura y de escritura del contenido de una celda.

Gestión disco. Con este tipo de comando se obtiene la memorización en disco de todo el tablero o de una parte del mismo.
Los ficheros previstos son dos, el primero para la memorización del contenido de las celdas, o sea de los buffers B$(*), el segundo para las fórmulas (F$(*)).
La función se desarrolla mediante dos subrutinas, la primera para la definición de la parte de tablero a memorizar y la segunda para la transferencia al disco.

La subrutina de transferencias utiliza dos pares de punteros:

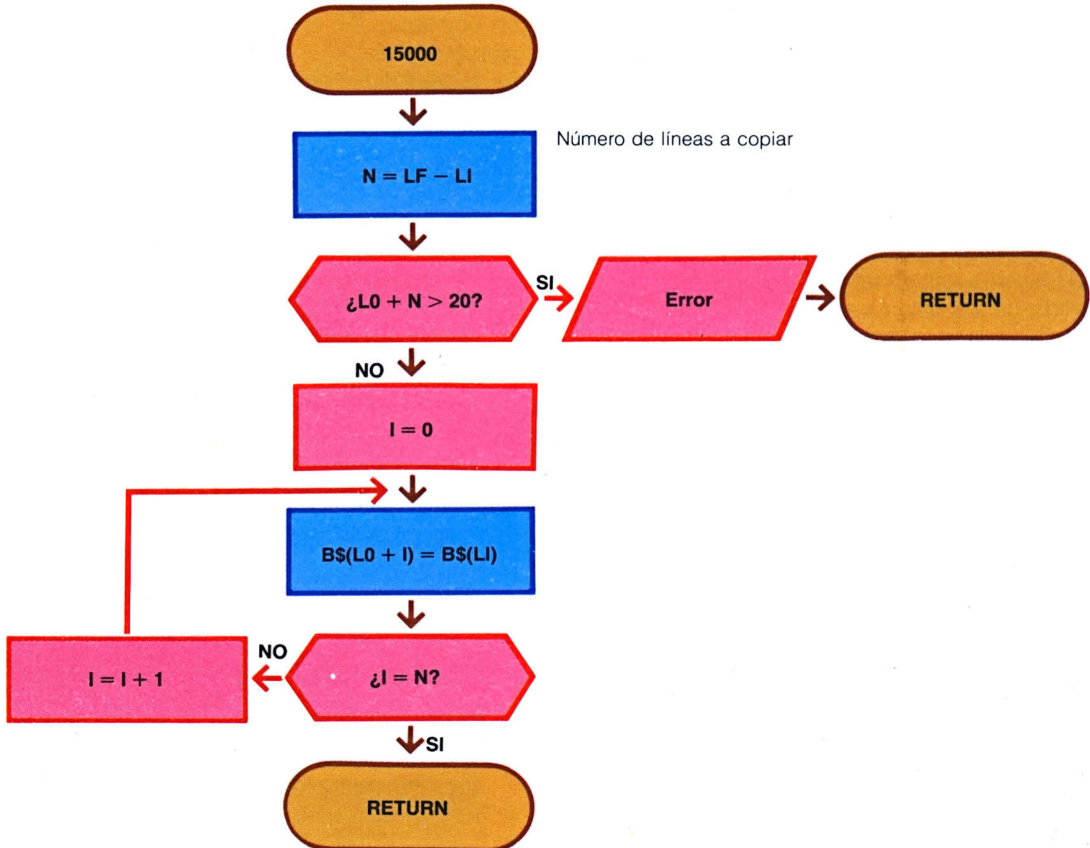
LI,CI = línea y columna de principio de memorización
LF,CF = línea y columna de final

implantados con una subrutina llamada anteriormente; por lo tanto, en la 16000 (gestión de comandos), para este tipo de comando, hay que insertar en sucesión dos subrutinas de llamada.
La primera se utiliza para la definición de la zona a memorizar (en la salida de los valores de LI,CI y LF,CF) y la segunda se emplea para la transferencia de estos valores al disco.
A la derecha se ha representado la subrutina de transferencia suponiendo una estructura que prevé como máximo 20 columnas y 20 líneas. Pueden obtenerse eventuales implantaciones simplemente variando los parámetros numéricos utilizados.
Además, la lógica empleada consiste en copiar en disco la zona de tablero indicada, dejando en blanco la zona que queda libre.
Esto implica que cada celda esté memorizada y representada teniendo en cuenta que hay que considerar siempre todas las direcciones como correspondientes a la posición del cursor. Con este procedimiento, se obtiene un posicionado diferente de las celdas en la fase de carga.
El diagrama de la subrutina para la determinación de los punteros LI,CI y LF,CF se ha omitido, porque se trata de una sencilla gestión del cursor o de una introducción directa (por el teclado), según si se desea pedir el margen indicándolo con el cursor o introduciéndolo.
Finalmente, téngase en cuenta que esta función no está en la versión de cinta.



SUBRUTINA DE COPIA POR LINEAS

En llamada:
LI,LF = Definen la parte a copiar
L0 = Indica la línea en la que empieza la copia

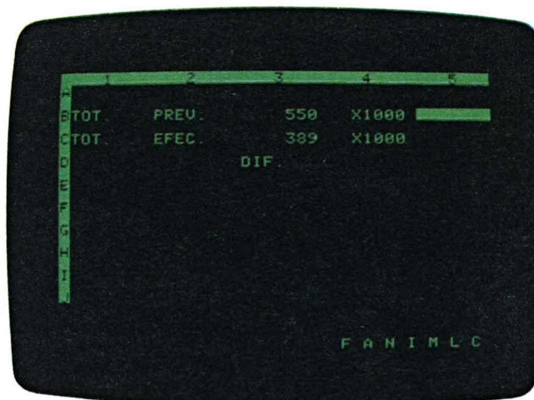


Comandos particulares. El tablero electrónico es un programa generalizado, incluso si se orienta al desarrollo de cálculos, por lo que debe incluir un cierto número de comandos que permitan su adaptación a las situaciones más dispares. El número y la variedad de dichos comandos dependen de la generalidad que se quiera dar al software. En esta versión se activa la única función que permite realizar una copia del contenido de las celdas de un punto a otro del tablero, pero pueden incluirse otras funciones, como la realización de los cálculos con sólo un comando, el posicionado del cursor con instrucciones del tipo GOTO, etc.
El diagrama de la subrutina para la copia del contenido de celdas puede verse arriba; su empleo requiere habitualmente la introducción de un código de activación, por ejemplo la letra "C" (Copia), la definición del margen a copiar y la indicación del punto inicial en que se posicionará la copia; por tanto, la lógica es análoga a la anterior. La función se realiza con dos subrutinas, la primera para la definición de la zona a copiar y de la dirección de partida de la copia. La segunda para la ejecución de la transferencia. El diagrama de la figura sólo se refiere a esta segunda subrutina y la primera puede reducirse a un simple coloquio que permita la introducción de:

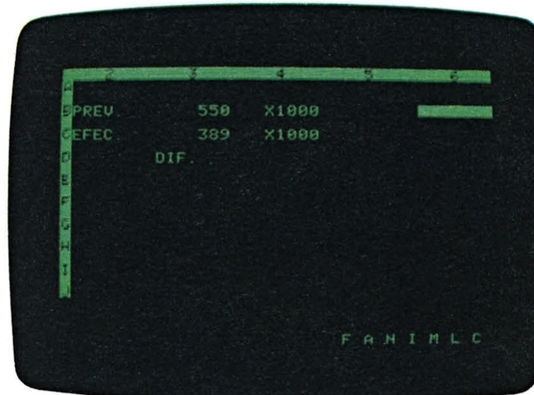
LI = Línea de principio parte a copiar
LF = Línea de final parte a copiar
LO = Línea en que empieza la copia

El ejemplo se realiza activando la única función de copia por líneas; si se desea por columnas, la lógica permanece la misma, variando los direccionados.
Además, obsérvese que el diagrama corresponde solamente a la transferencia del contenido de las celdas, mientras que las fórmulas permanecen sin variación.

Tablero electrónico. La versión DOS de este programa prevé, para el posicionado, las teclas:
CTRL + W: desplaza arriba
CTRL + A: desplaza a la izquierda
CTRL + S: desplaza a la derecha
CTRL + Z desplaza abajo
Los comandos utilizables (presentados en la parte inferior de la pantalla) son:
F = introducción fórmulas
A = introducción datos alfanuméricos
N = introducción datos numéricos
I = inserción línea o columna
M = memorización
L = carga desde el disco
C = copia



Moviendo el cursor más allá de los límites de la pantalla de base (delimitada por las casillas A5 y J1) se produce el deslizamiento de los datos presentados; este mecanismo permite aprovechar toda el área de trabajo ofrecida por el tablero (15×26 casillas de introducción). En la foto de al lado se muestra el deslizamiento de los datos hacia la izquierda: por tanto, se está moviendo el indicador de casilla hacia la derecha.



Introducción fórmulas. El comando F permite la introducción de una fórmula matemática en la casilla sobre la que está posicionado el cursor. En este caso, la casilla considerada está identificada por las coordenadas D4. El cálculo a efectuar es la diferencia entre los valores numéricos contenidos en las casillas B3 y C3.



Cálculo. Apenas introducida la fórmula, el programa realiza la sustitución de los valores numéricos contenidos en la casilla interesada y el desarrollo del cálculo. El resultado así obtenido se presenta en la casilla que contiene la fórmula originaria.



Introducción datos: Los comandos A y N permiten la introducción de datos, respectivamente alfanuméricos y numéricos, en el área de trabajo del tablero. En este caso, se está insertando el valor 450 en la casilla C3, en la que hay un valor numérico.

Una vez modificado el valor numérico de una casilla contenido en un cálculo, se puede obtener la actualización del resultado de éste posicionando sencillamente el indicador en la casilla que contiene la fórmula. Efectivamente, en la memoria no está contenido el resultado (que se lee en las coordenadas D4), sino la fórmula en la misma forma en que se ha insertado.

Inserción. Se ha activado el comando I; la primera pregunta del programa es sobre el tipo de inserción a realizar:

L = inserción línea

C = inserción columna

Con la tecla E puede salirse de la opción, anulando el efecto.

La inserción se ha realizado añadiendo una línea al tablero. El efecto del comando es que todos los datos que antes ocupaban las casillas a partir de la línea C han sido trasladados hacia abajo y ahora están a partir de las coordenadas D1. La línea C está ahora disponible para ulteriores introducciones.

Tablero electrónico. La versión MSX tiene algunas particularidades, como la forma del indicador de casilla ocupada que, en este caso, al faltar una gestión fácil de los caracteres REVERSE, está representado por el cursor normal.
Los comandos previstos son:
F = introducción fórmulas
A = introducción datos alfanuméricos
N = introducción datos numéricos
I = inserciones
C = copia
Para el desplazamiento de cursor se tiene:
→ = desplaza a la derecha
← = desplaza a la izquierda
↑ = desplaza arriba
↓ = desplaza abajo

Introducción fórmulas. Se ha activado el comando F: en la parte inferior derecha de la pantalla se presenta la tecla correspondiente. En este caso se ha introducido el cálculo de la diferencia entre las casillas C4 y C3, cuyo resultado se ha presentado en las coordenadas C5.

En la foto de al lado se ha presentado la parte de tablero que inicialmente no se muestra. La pantalla sobre la que empieza el trabajo (después del comando RUN) está delimitada por las casillas A1-J5: llevando el cursor hacia abajo, se obtiene la presentación de las casillas F1 a O5.

Introducción datos numéricos. Mediante el comando N es posible insertar en el tablero valores numéricos. Es importante distinguir este tipo de introducción de las otras: efectivamente, introduciendo 1500 con el comando A, su valor será 0, porque se ha insertado como alfanumérico (en forma de comando o descripción).

Inserción. El comando I permite la inserción de líneas o columnas en una zona del tablero. En la parte inferior izquierda de la pantalla se presentan las posibles opciones:

L = inserción línea
C = inserción columna
E = salida del comando

Se ha activado el comando de inserción de columna. Todos los datos presentados en las casillas de la columna 2 se han trasladado a la siguiente. Ahora, la columna 2 está libre para nuevas introducciones.

Comando C (copia). Este comando permite representar una parte del tablero ya definida en otra todavía libre. Los primeros parámetros a proporcionar para esta operación son la dirección de partida y la de llegada de la zona a copiar. Por ejemplo, si se quieren duplicar las dos primeras líneas, tendremos: A1-B26.

La siguiente introducción para activar el comando de copia es la de la dirección en que hay que empezar a duplicar datos.

Extensiones y modificaciones

Por tratarse de un programa de empleo generalizado, las posibles modificaciones e implantaciones son innumerables y consisten, principalmente, en la introducción de funciones de alto nivel que contienen diferentes instrucciones elementales para evitar al usuario la necesidad de desarrollar el detalle.
La estructura utilizada en este programa es de tipo abierto, por lo que son posibles todas las variaciones necesarias para adaptar mejor el software a las necesidades específicas.
Sin embargo, hay que observar que el empleo del lenguaje Basic, si bien por una parte hace mucho más ágil la programación, en cambio implica una limitación del área de memoria direccionable, por lo que para aplicaciones particularmente complejas es necesario prever subrutinas en Assembler que direccionen zonas de memoria externa al ambiente Basic.
A continuación se presenta un ejemplo correspondiente al cálculo de gastos comunitarios.

Ejemplo de aplicación. En la gestión de una propiedad comunitaria, cada gasto se reparte entre los copropietarios en base a una tabla de milésimas, que expresa la cuota a pagar por cada uno. En general, las tablas para la gestión de una comunidad son más de una, relativas cada una a un cierto tipo de gasto. Por ejemplo, la correspondiente a los consumos de combustible para la calefacción se obtiene de la superficie de los radiadores que tiene cada copropietario.
Para fijar mejor las ideas, la siguiente aplicación se ha desarrollado según este tipo de gasto; sin embargo, también es válida para otras aplicaciones, aunque naturalmente varían los valores y los respectivos importes. Supongamos que la propiedad comunitaria está constituida por cinco apartamentos, cada uno con las siguientes superficies radiantes:

Piso 1: 11,96 metros cuadrados
Piso 2: 3,50 metros cuadrados
Piso 3: 11,27 metros cuadrados
Piso 4: 9,98 metros cuadrados
Piso 5: 12,93 metros cuadrados

Después de haber introducido los valores (comando N, introducción de datos numéricos), se calcula el total de la superficie radiante en la casilla identificada por las coordenadas H2. Para ello se utiliza la función SUM (ver la primera foto de la pág. 2418).
En base a las superficies instaladas hay que calcular la tabla de distribución milesimal, que se utilizará para obtener los coeficientes de atribución de los gastos.
Esta tabla expresa el porcentaje que grava sobre cada apartamento expresado en milésimas para hacer mejores aproximaciones de los cálculos. El cálculo a realizar para obtener el coeficiente de cada apartamento es:

$$\text{milésimas} = \frac{\text{superficie} \times 1000}{\text{total}}$$

Por ejemplo, en el piso 1 tendremos:

$$\text{milésimas} = \frac{11{,}96 \times 1000}{49{,}64} = 241 \text{ (aprox.)}$$

En el tablero, este cálculo puede realizarse fácilmente implantando la fórmula:

$$\text{celda de milésimas} = \frac{1000}{49{,}64} \times \text{celda de superficie}$$

Por ejemplo (ver la foto de la pág. 2418), suponiendo que se han introducido previamente las superficies de los cinco apartamentos en las celdas identificadas por los cruces entre la columna dos y las líneas B, C, D, E, F, el cálculo será:

- en la celda B4 (milésimas piso 1) = (1000/H2)*B2 (la superficie está en B2)
- en la celda C4 (milésimas piso 2) = (1000/H2)*C2 (la superficie está en C2)

y así sucesivamente para las otras.
Finalizados estos cálculos, se dispone en la columna 4 (y por tanto en las celdas B4 y F4, ver la tercera foto de la pág. 2418) de las milésimas correspondientes a cada propiedad comunitaria.
Para obtener la distribución de un gasto hay que dividirlo primero por 1000 y después multiplicarlo por las milésimas de cada piso. Por tanto, tendremos la siguiente fórmula:

$$\text{importe de cada piso} = \frac{\text{gasto total}}{1000} \times \text{milésimas}$$

En la última foto de la página 474 se ha representado este último paso; el gasto total se ha introducido en la casilla J3, por lo que las fórmulas serán:

K3 (importe primer piso) = (J3/1000)*B4
L3 (importe segundo piso) = (J3/1000)*C4

y análogas para las restantes. En este punto, en la columna 3 y de la línea K a la O se han representado los gastos de los pisos individuales.

La Base de Datos

El empleo más frecuente de los procesadores electrónicos es en la gestión de archivos. Analizando las posibles aplicaciones del ordenador, se hace evidente que una parte importante de los programas corresponde a la memorización y a la búsqueda de los datos, o sea a la gestión de un archivo.
La Base de Datos permite el desarrollo de estas aplicaciones sin necesidad de ningún conocimiento de programación.
La aplicación principal del programa, como ya se ha indicado, es para la memorización, la búsqueda y la impresión de datos; por tanto, constituye el complemento indispensable del tablero electrónico.
Los sectores de aplicación variados: desde el listín telefónico a una relación de Clientes y Proveedores, o a la gestión automática de una biblioteca.
Además, la versión presentada contiene un módulo dedicado al desarrollo de fórmulas. Por tanto, es posible integrar determinadas funciones de cálculo, típicas del tablero, en la gestión de los archivos.

Uso del programa

La gestión de un archivo requiere, como mínimo, las siguientes funciones:

- Creación
- Introducción datos
- Actualización
- Búsqueda
- Impresión

Las distintas funciones se explicitan en otras tantas subrutinas (o programas) que, si bien tienen los significados lógicos usuales, se realizan con nuevas funcionalidades y están orientadas al objetivo particular del programa.
La gestión de un archivo con la ayuda de un ordenador presupone la necesidad de memorizar y procesar una gran masa de datos, por lo que es necesario disponer de un programa que pueda adaptarse no sólo a las necesidades, puramente estéticas, del usuario, sino que también ofrezca una buena flexibilidad de empleo. En particular es necesario recurrir a una Data Entry completamente parametrizada, que pueda utilizarse y modificarse sin entrar en los detalles de la programación.
La Data Entry utilizada hasta ahora sólo satisface parcialmente estas necesidades: de hecho, los elementos típicos de la máscara vídeo (posición, longitud y tipo de campos, etc.) sólo pueden modificarse mediante instrucciones DATA. Por tanto, su modificación requiere el conocimiento, por lo menos superficial, del lenguaje de programación. Una alternativa podría ser la de utilizar una máscara predefinida, no accesible al usuario, dedicada a la introducción de los parámetros característicos de los campos; sin embargo, se trata de una solución que requiere la participación del usuario limitándolo a «interpretar» la estructura final que asumirá la máscara. Para eliminar el inconveniente se ha introducido una nueva subrutina que permite generar las máscaras vídeo «dibujándolas» en la pantalla.
Desde el punto de vista funcional, la operación se realiza como sigue: el usuario activa la voz de generación y el sistema presenta una pantalla vacía, con sólo las indicaciones de línea y de columna en que se encuentra el cursor; en este momento, el usuario puede posicionar en una zona cualquiera de la pantalla (el posicionado en términos de línea y columna se presenta siempre) y definir en ella una descripción o un campo de datos. En el primer caso, los caracteres que constituyen la descripción deben ir precedidos y seguidos por el símbolo «&» (que después se eliminará). En el segundo caso (zona de datos), los caracteres introducidos son:

"X" = define un campo de tipo alfanumérico
"N" = para los campos numéricos
"C" = define un campo como resultado de cálculo
"." = delimita la parte entera de un campo numérico.

Al final de las definiciones de la máscara, con la tecla de introducción se activa el proceso de lectura y memorización de la pantalla. De los valores contenidos en ella, el sistema toma los parámetros característicos de la máscara y del fichero asociado.
En el caso en el que algunos campos se hayan definido como de cálculo, el sistema procede a pedir la fórmula de desarrollo. Esta fórmula puede contener:

- Datos numéricos
- Símbolos de las operaciones aritméticas
- Referencias a otros campos
- Paréntesis, incluso sobre varios niveles
- Funciones internas

La simbología a adoptar es la utilizada normalmente, excepto las referencias a otros campos, que se obtienen escribiendo el número del campo entre comillas ("6" se refiere al campo número 6). Por ejemplo, si se desea construir una máscara vídeo con:

1 / Nombre
2 / Importe
3 / Porcentaje a aplicar al importe
4 / Importe correspondiente al porcentaje

Las operaciones a realizar son:

- Posicionar el cursor en la zona de pantalla en la que se desea la descripción de cada campo
- Escribir la descripción, precedida y seguida del símbolo «&»
- Posicionar el cursor en la zona vídeo en la que se desea el campo de datos
- Introducir las características del área de datos utilizando la simbología anterior, o sea:

  — El campo Nombre estará representado por un número de carácteres "X" igual a la longitud máxima que se desea dar al campo
  — El importe y el porcentaje (campos numéricos en introducción) deberán definirse con símbolos "N" (siempre en un número igual a los bytes reservados al campo correspondiente)
  — El porcentaje deberá definirse como resultado de cálculo, o sea con el símbolo "C".

Por ejemplo, la máscara vídeo terminada podrá tener la forma:

& Nombre & XXXXXXXXXX	alfanumérico, 10 caracteres
& Importe & NNNNN	numérico, 5 cifras
& Porcent. & NN.N&%&	numérico, 2 enteros 1 decimal
& Valor % & CCCCC.C	de cálculo, 5 enteros, 1 decimal

El último paso consiste en definir la fórmula para el campo del cálculo:

Valor % = Importe × Porcentaje/100

Por tanto, la fórmula correspondiente es:

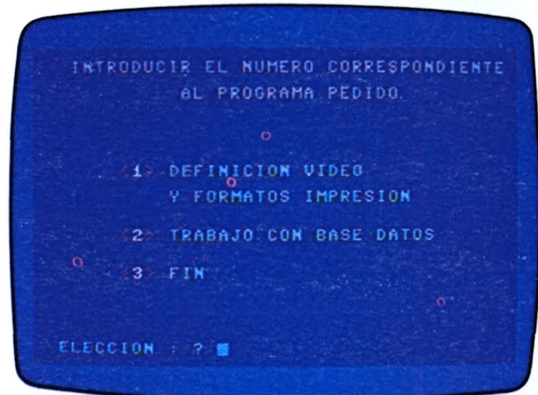
"2"*"3"/100

donde los dos primeros números (2 y 3) encerrados entre comillas identifican respectivamente los campos número 2 (Importe) y número 3 (Porcentaje). El tercer número (100) se supone un valor numérico porque no está encerrado entre paréntesis.
Terminadas las definiciones, puede procederse a la introducción de los datos. La máscara vídeo que aparece reclamando la función es idéntica a la definida por el usuario, con los símbolos "&" eliminados y con las zonas de datos indicadas con la simbología usual, o sea con tantos símbolos "–" como cuantos bytes haya previstos en los campos alfanuméricos, y con símbolos "#" en un número que corresponda a las cifras previstas en los campos numéricos.
Esta particular gestión de máscara vídeo ofrece la posibilidad de trabajar de manera muy expedita en su construcción, siempre que no se memorice ningún parámetro, procediendo a su memorización sólo después de haberla ultimado. Además, es muy sencillo aportar correcciones, porque los datos, o sea los parámetros de los distintos campos, hasta el momento de la confirmación (con tecla funcional) están simplemente escritos en la pantalla y aún no memorizados, es decir, no han alterado ninguno de los parámetros que definen la máscara. Las eventuales correcciones se hacen en la memoria de vídeo y no necesitan ninguna lógica ni modificación de los parámetros. Como se ha indicado, el final de la preparación, la adquisición de los parámetros se hace leyendo la memoria vídeo y tomando de su contenido los valores a asignar a la máscara en fase de construcción, que son los mismos utilizados en la Data Entry anterior, o sea:

- Posición de la descripción (línea y columna)
- Descripción
- Tipo de campo
- Longitud
- Número de cifras decimales
- Posición del campo (línea y columna)

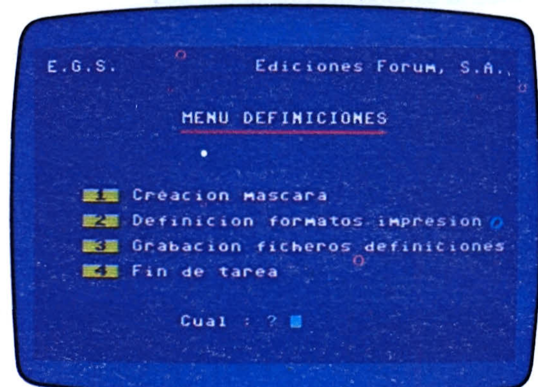
Todos estos valores pueden obtenerse leyendo la memoria vídeo y teniendo en cuenta que las descripciones empiezan y terminan con el símbolo "&", mientras que las definiciones se representan con uno de los símbolos "X", "N" o "C".
La rutina que realiza esta transferencia (de imagen a valores) consiste en un bucle que explora el vídeo (o sea la zona de memoria correspondiente) y después de haber determinado si el contenido de una localización es un campo o una descripción, toma de aquél las características. La elección de una u otra posibilidad (descripción o campo) se toma en base a la presencia o la ausencia del símbolo "&". En el primer caso, todos los caracteres que siguen, hasta el siguiente "&", son considerados como pertenecientes a la descripción y, por tanto, son memorizados en la matriz D$(*). Y viceversa, si se encuentra un carácter sin que se haya activado el flag de descripción, el contenido de la pantalla es interpretado como una definición, y se cuentan los símbolos (C, X, N) para determinar la longitud del campo (LC(*)).
En ambos casos (descripción o campo de datos), las coordenadas son determinadas por el propio bucle de exploración de la pantalla. La única precaución a tomar al generar una máscara vídeo consiste en no utilizar descripciones de longitud superior a 10 caracteres (los sobrantes son eliminados) y en limitar el número de campos de datos a 20, que es el valor máximo previsto. Estas limitaciones pueden superarse fácilmente aumentando los dimensionados.

Base de Datos. La foto muestra el menú inicial de acceso al programa. La primera función a activar es la voz 1 (Definiciones).

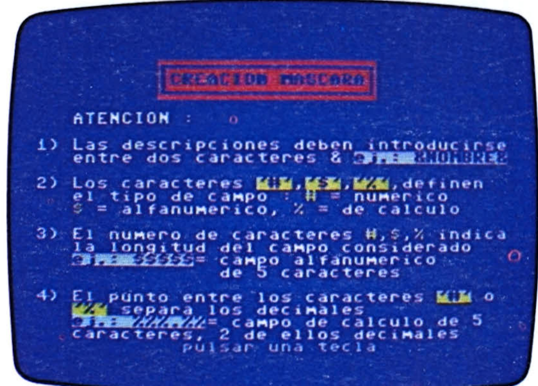


Menú definiciones. Activando esta función aparece el menú secundario para la selección de la subvoz. Las previstas son:

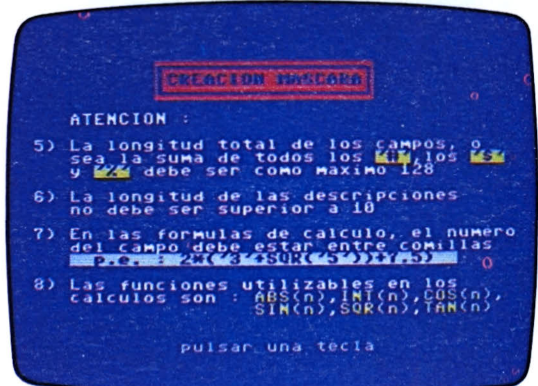
1. Creación máscara: define la máscara vídeo, el tipo de los campos y los eventuales cálculos.
2. Definición formatos de impresión: permite implantar la forma que deberán tener los tabulados y los campos que se desea enviar a impresión.
3. Grabación ficheros definiciones: utilizada para memorizar en disco los valores de definición introducidos con las voces anteriores.



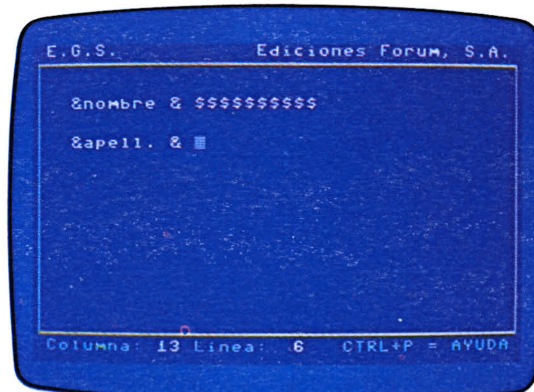
Creación máscara. La foto muestra las instrucciones necesarias para la utilización de la máscara. Constituye un recordatorio que evita al usuario consultas continuas del texto.



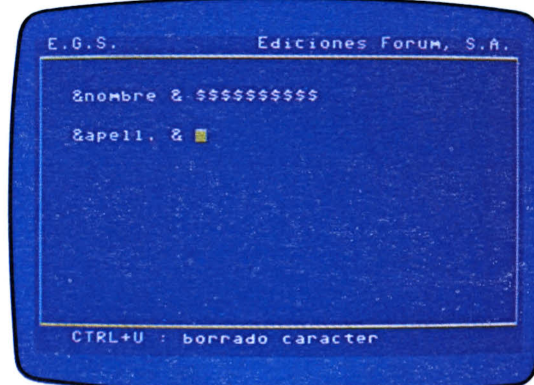
Segunda parte de las instrucciones: en ella aparece también la relación de las funciones reconocidas que después podrán emplearse en el desarrollo de los cálculos.



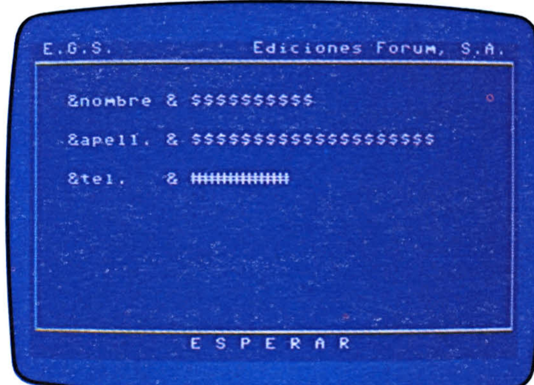
Ejemplo de definición. Los símbolos & definen las zonas vídeo reservadas a las descripciones de los campos, mientras que el símbolo $ se utiliza para indicar la longitud de los campos alfanuméricos. En la parte inferior, última línea, aparece la posición actual del cursor.



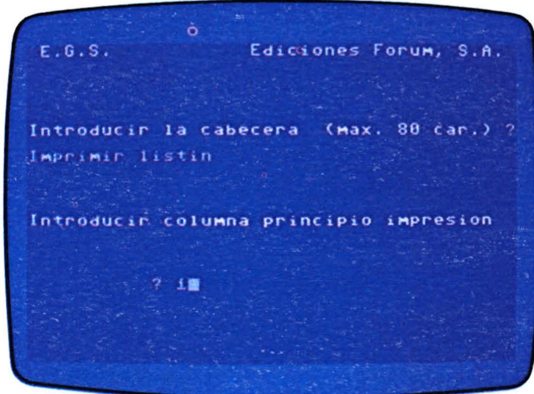
La foto muestra el empleo de la línea de coloquio. Activando la función ayuda en esta línea se presentan los comandos previstos.



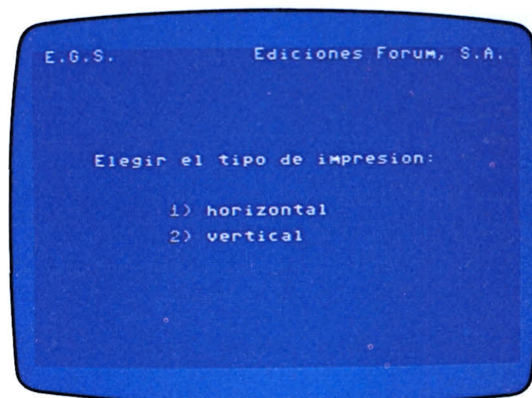
La máscara vídeo de la foto anterior se ha completado. Los campos previstos son 3: los dos primeros alfanuméricos y el tercero numérico.



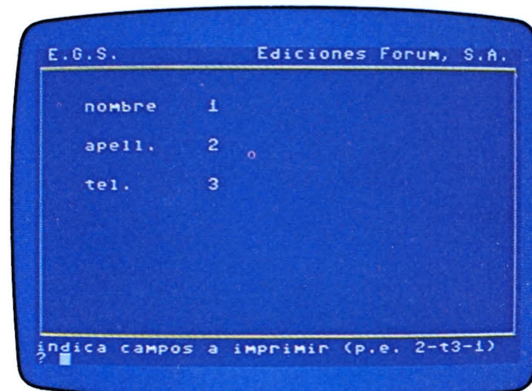
Activando la voz 2 del submenú de generación se tiene la presentación del coloquio para la definición de la impresión. La foto muestra la primera pregunta.



A continuación, el usuario debe seleccionar el tipo de impresión. Con la voz 1, los campos seleccionados son presentados en una misma línea y con la voz 2, uno por línea.



La última fase de la impresión consiste en la selección de los campos.
El sistema presenta todos los previstos con un número progresivo al lado. El usuario debe introducir los números correspondientes a los campos a imprimir. En esta fase, la letra "t" indica que el contenido del campo correspondiente se utiliza para el cálculo del total.



Fase de memorización: el sistema espera un acuerdo antes de iniciar la operación de escritura en el disco.



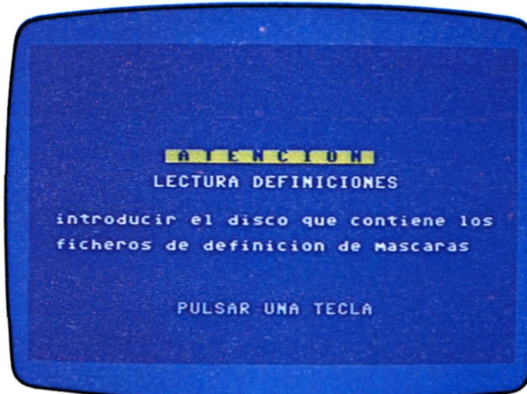
Durante la escritura en el disco, la máquina presenta eventuales necesidades de sustitución del soporte. La foto muestra la fase final de las definiciones.



Terminada la fase de definición puede pasarse a la de utilización, seleccionando la voz 2 del menú principal. La foto muestra el menú secundario correspondiente a este procedimiento.



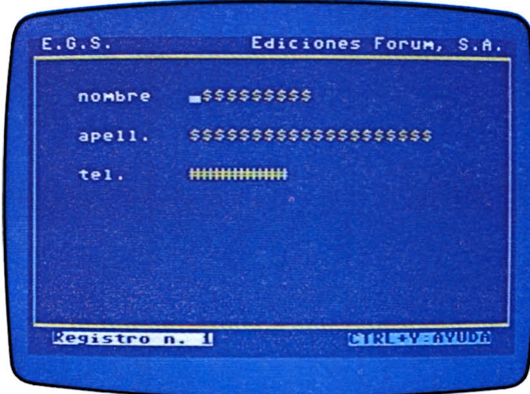
Antes de cualquier otra operación, el sistema debe cargar las definiciones. Como de costumbre, aparece la petición de introducir el diskette.



Terminada la carga de las definiciones, el programa pregunta si el fichero que deberá utilizarse existe o debe crearse.



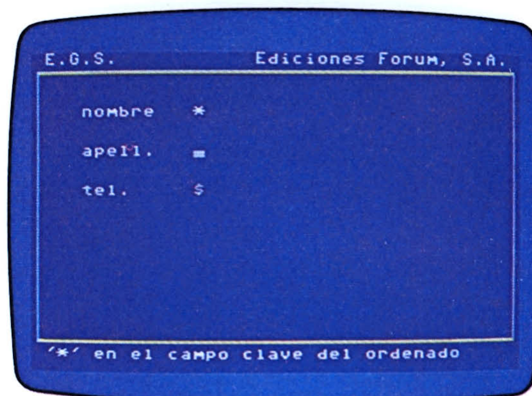
La foto muestra la máscara vídeo de introducción de datos consiguiente a las anteriores definiciones.



Terminadas las introducciones, el programa pasa a la memorización de los datos en el disco.



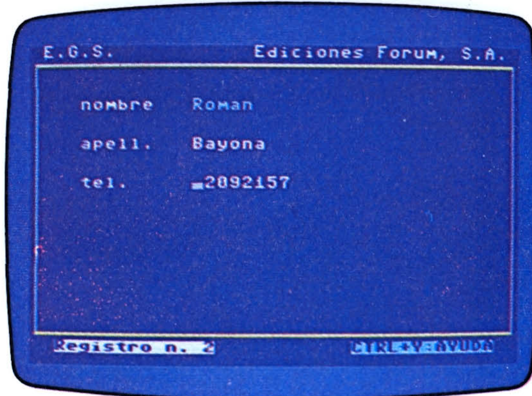
Máscara vídeo para el ordenado. En esta función (ordenado de los datos) se presentan todos los campos que forman el registro, poniendo de manifiesto el elegido como clave de ordenado con el símbolo '*'.



Terminado el ordenado puede procederse a la búsqueda. La foto muestra la fase inicial de esta función.



El usuario ha pedido el nombre "Román", que está memorizado en el registro n.° 2. El sistema lo selecciona (en base al campo "nombre") y presenta su contenido.





Base de Datos/Versión Commodore 64

Cargador

```
10 REM ************************
20 REM *   BASE DATOS C-64    *
30 REM ************************
35 POKE53280,14:POKE53281,6:POKE650,0:PRINTCHR$(142)CHR$(8)
40 PRINT"":PRINT"INTRODUCIR EL NUMERO CORRESPONDIENTE"
42 PRINT"AL PROGRAMA PEDIDO"
44 PRINT"1 DEFINICION VIDEO"
45 PRINT"FORMATOS IMPRESION"
46 PRINT"2 TRABAJO CON BASE DATOS"
47 PRINT"3 FIN "
50 INPUT"ELECCION : ";R
55 IF R<1 OR R>3 THEN 40
56 IF R=3 THEN PRINT"":END
57 PRINT""SPC(4)"ESPERAR LA CARGA, POR FAVOR"
60 ON R GOTO 100,200
100 PRINT"POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
110 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"D1"CHR$(34)",8"
120 GOTO1000
200 PRINT"POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
210 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"D2"CHR$(34)",8"
1000 POKE198,10:FORL=631TO640:READN:POKEL,N:NEXT
1010 DATA19,13,145,145,13,82,85,78,58,13
1050 REM :
1051 REM    D1 = GENERACION
1052 REM
1053 REM    D2 = IMPRESION
```

Definiciones

```
39999 REM
40000 REM PROGRAMA PARA :
40001 REM DEFINICION MASCARA VIDEO
40002 REM Y FORMATOS IMPRESION
40003 REM
40030 HO$="":POKE 650,128:PRINTHO$;CHR$(14);
40035 M3=20
40040 DIM D$(M3),TP(M3),LC(M3),ND(M3),XD(M3),YD(M3),XC(M3),YC(M3)
40045 DIM FO$(M3),FT$(M3),NS$(M3)
40050 SP$="":FORI=1TO40:SP$=SP$+CHR$(32):NEXT I:S1$=SP$+SP$
40070 H1$=" ¯.|.♥.             ¯DICIONES _ORUM, ♥.♠."
40200 REM ********
40201 REM * MAIN *
40203 REM ********
40210 CLOSE1:CLOSE2:CLOSE15
40230 GOSUB 46000
40240 ON OPZ GOSUB 41000,45000,42500
40250 GOTO 40200
41000 GOSUB43500:PRINTHO$:POKE53280,14
41010 X=1:Y=2:GOSUB 41150:GOSUB 41020:GOTO 41070
41020 A$="−":FOR I=1 TO 38:L$=L$+A$:NEXT I
41030 L1$="┌"+L$+"┐":L2$="└"+L$+"┘":L$=""
41040 X1=1:Y1=0:GOSUB61000:PRINT""H1$;"";
41050 POKE646,7:PRINTL1$;:FOR I=1 TO 20:PRINT"|"+LEFT$(SP$,38)+"|";:NEXT:PRINTL2
$;
41055 PRINT "";
41060 RETURN
41070 REM
41071 REM GENERACION
41072 REM
41090 FOR I=1 TO M3:LC(I)=0:TP(I)=0:ND(I)=0:NEXT I
41100 GOSUB 41310
41110 A=ASC(I$)
41111 IF A=58 OR A=44 OR A=59 OR A=0 THEN 41190
41112 IF A<32 THEN 41180
41120 IF A>95 THEN 41100
41130 GOSUB60000:PRINT""I$:X=X+1:IF X>38 THEN X=38
41132 GOSUB 41150
41140 GOTO 41100
41150 REM
41160 J1$="":J2$="":X1=1:Y1=23:GOSUB61000:PRINTSP$;:IFY<10THENJ2$=CHR$(32)
41161 IFX<10THENJ1$=CHR$(32)
41162 GOSUB61000:PRINT"COLUMNA:";J1$;X;TAB(7)"LINEA:";J2$;Y-1;
41163 PRINT"  ┤ ┴+┐ = ♠|╱♠";
41170 RETURN
41180 REM
41185 REM COMANDOS
41190 REM
41200 REM
41210 IF A=19 THEN X=X+1:X2=X-1:Y2=Y:GOSUB55000:IF X>38 THEN X=1:A=26:GOTO 41240
41220 IF A=1 THEN X=X-1:X2=X+1:Y2=Y:GOSUB55000:IF X<1 THEN X=38:A=23
41230 IF A=23 THEN Y=Y-1:X2=X:Y2=Y+1:GOSUB55000:IF Y<2 THEN Y=2
41240 IF A=26 THEN Y=Y+1:X2=X:Y2=Y-1:GOSUB55000:IF Y>21 THEN Y=21
41250 IF A=13 THEN X2=X:Y2=Y:GOSUB 55000:X=1:Y=Y+1:IF Y>21 THEN Y=21
41260 IF A=9 THEN GOSUB 41360
41270 IF A=21 THEN GOSUB 41400
41280 IF A=15 THEN X2=X:Y2=Y:GOSUB55000:GOTO41460
41290 IF A=16 THEN GOSUB 44000
41300 GOSUB 41150:GOTO 41100
41310 REM ******* CARACTER *******
41320 GET I$:IF I$<>"" THEN RETURN
41321 IF I$="" THEN 41323
41322 IFASC(I$)=58ORASC(I$)=59ORASC(I$)=44THEN 41320
41323 LC=1024+(40*Y)+X:CH=PEEK(LC)
41324 POKE LC,CH
41326 FOR T=1 TO 40:NEXT T
41328 IF CH>128 THEN POKE LC,CH-128:GOTO 41320
41330 POKE LC,CH+128
41332 GOTO 41320
41360 REM INSERT
41362 P1=0:P2=Y*40+X:P3=(Y+1)*40-2
41365 FOR I=P3 TO P2STEP-1
41366 X=38-P1:L1=1024+(I-1):L2=1024+(Y*40)+X:CH=PEEK(L1)
41367 IF CH=160 THEN CH=32:GOTO 41369
41368 IF CH>128 THEN CH=CH-128
41369 POKE L2,CH
41370 P1=P1+1:NEXT I:GOSUB60000:PRINT" ";
41390 RETURN
41400 REM BORRADO
41405 XF=X:P1=0:P2=Y*40+X:P3=(Y+1)*40-2
41410 FOR I=P2 TO P3
41415 X=XF+P1:L1=1024+(I+1):L2=1024+(Y*40)+X:CH=PEEK(L1)
41420 IF CH=160 THEN CH=32:GOTO 41425
41422 IF CH>128 THEN CH=CH-128
41425 POKE L2,CH
41430 P1=P1+1:NEXT I:GOSUB60000:PRINT" ";
41435 X=XF:RETURN
41460 REM
41461 REM LECTURA MASCARA VIDEO
41462 REM
41465 X1=0:Y1=23:GOSUB61000:PRINTLEFT$(SP$,13);"┌ ♥ ┐ ¯ _ ♠ _";LEFT$(SP$,13);
41470 REM
41500 YY=1:PD=0:PC=0
41510 FOR JJ=2 TO 21:J=JJ*40:YY=YY+1
41520 FOR K=1 TO 38
41530 CH=PEEK(1024+(K+J)):GOSUB 62000
41535 REM
41540 IF CH=38 THEN 41660
41550 IF CH=35 OR CH=36 OR CH=37 THEN 41570
41560 IF CH<>32 THEN 41770
41562 GOTO 41730
41570 PC=PC+1:IF PC>M3 THEN 41810
41580 IF CH=36 THEN TP(PC)=3:GOTO 41590
41582 IF CH=35 THEN TP(PC)=1:GOTO 41590
41584 TP(PC)=2
41590 LC(PC)=1:YC(PC)=YY:XC(PC)=K:ND(PC)=0:W6=0
41600 K=K+1:IF K=40 THEN 41770
41610 IF W6=1 AND CH=36 THEN 41770
41620 IF PEEK(1024+(J+K))=CH THEN LC(PC)=LC(PC)+1:GOTO 41622
41621 GOTO 41630
41622 IF W6=1 THEN ND(PC)=ND(PC)+1
41624 GOTO 41600
```

```
41630 V=PEEK(1024+(J+K)):IF V=160 THEN V=32:GOTO 41635
41632 IF V>128 THEN V=V-128
41635 IF V=46 AND W6=0 THEN LC(PC)=LC(PC)+1:W6=1:GOTO 41600
41640 IF V=32 OR V=110 OR V=93 OR V=125 THEN K=K-1:GOTO 41730
41650 IF V<>38 THEN 41770
41660 REM DEFINICIONES
41670 PD=PD+1:IF PD>M3 THEN 41810
41680 D$(PD)="":XD(PD)=K+1:YD(PD)=YY
41690 K=K+1:IF K=39 THEN 41770
41700 KH$=CHR$(PEEK(1024+(K+J))):A1=ASC(KH$):IF A1<32 THENA1=A1+64:KH$=CHR$(A1)
41702 IF KH$="&" THEN 41730
41710 D$(PD)=D$(PD)+KH$
41720 GOTO 41690
41730 NEXT K:NEXT JJ
41740 AA=0:FOR I=1 TO M3:AA=AA+LC(I):NEXT I
41750 IFAA>128THENH$="LONG. CAMPOS > 128":GOTO 41820
41760 GOTO 41850
41770 REM ******* ERROR *******
41800 H$="ERROR EN LA LINEA"+STR$(YY-1):GOTO 41820
41810 H$="CAMPOS O DESCR. >"+STR$(M3)
41820 X1=1:Y1=23:GOSUB61000:PRINTH$;" : CORREGIR ";:FOR T=1 TO 4000:NEXTT
41830 X1=0:Y1=23:GOSUB61000:PRINTLEFT$(SP$,38);:X=1:Y=YY:GOSUB 41150:GOTO 41070
41850 REM *** DEFINICION CALCULOS ***
41860 REM
41880 MX=PC:NC=0
41890 FOR I=1 TO MX
41900 IF TP(I)=3 THEN 41930
41910 X1=XC(I):Y1=YC(I):GOSUB61000:PRINTLEFT$(SP$,LC(I))
41920 GOSUB61000:PRINTSTR$(I);
41930 NEXT I
41940 X1=0:Y1=23:GOSUB61000:PRINTLEFT$(SP$,39);
41950 FOR I=1 TO MX:FO$(I)=""
41960 IF TP(I)<>2 THEN 42070
41970 X1=3:Y1=23:GOSUB61000:PRINT"INTRODUCIR LA FORMULA DEL CAMPO";I;
41972 Y=24:X=4
41990 X1=3:Y1=24:GOSUB61000:PRINT"▯>";:X1=35:GOSUB61000:PRINT"▯<▯";
41990 GOSUB41310:IF I$=CHR$(1) THEN X=X-1:IF X<4 THEN X=4
42000 IF I$=CHR$(19) THEN X=X+1:IF X>34 THEN X=34
42010 IF I$=CHR$(13) THEN 42050
42020 IF ASC(I$)<32ORASC(I$)>95 THEN 41990
42030 GOSUB60000:PRINT"▯"I$;:X=X+1:IF X>34 THEN X=34
42040 GOTO 41990
42050 FOR H=1988 TO 2018
42060 CH=PEEK(H)
42062 GOSUB 62000
42067 FO$(I)=FO$(I)+CHR$(CH):NEXT H
42068 X1=3:Y1=24:GOSUB61000:PRINTLEFT$(SP$,36);
42070 NEXT I
42080 FOR I=1 TO 1000:NEXT I:W5=2:RETURN
42500 REM
42501 REM MEMORIZACION MASCARA
42502 REM
42505 PRINTHO$:X1=12:Y1=6:GOSUB61000:PRINT"▯▯ | ‾ / ‾ ˎ ┌ /"
42507 X1=5:Y1=12:GOSUB61000
42510 PRINT"▯ˎ/┘‾┌‾⌒ ‾/ L▯ ´/ˎ‾▯‾´ ‾ˎ▯‾┌"
42512 X1=3:Y1=Y1+2:GOSUB61000:PRINT"▯┌L ▯┌‾┬┘‾ ┘‾▯‾x▯‾┬ ▯ ‾‾ˎ/ˎ‾ˎ┌/‾▯"
42514 X1=11:Y1=20:GOSUB61000:PRINT"▯┐L▯▯‾ ´/▯ |‾‾L▯":POKE 198,0:WAIT 198,1
42520 X1=0:Y1=20:GOSUB61000:PRINTSP$;
42522 X1=3:GOSUB61000:PRINT"▯HAY QUE FORMATEAR EL DISCO ? (▯▯▯/▯▯/▯▯)"
42524 GET RI$:IF RI$<>"S" AND RI$<>"N" THEN 42524
42526 IF RI$="N" THEN 42550
42528 X1=0:Y1=20:GOSUB61000:PRINTSP$;
42529 X1=9:GOSUB61000:PRINT"▯▯ ‾ ┌ ‾ ˎ ▯ | ‾ ▯ ‾ ┌ "
42530 OPEN15,8,15,"N0:BASIC ADVANCED,01":PRINT#15,"I":CLOSE15
42550 OPEN1,8,15,"I/O":GOSUB63000
42552 CLOSE1:IF ER<>0 THEN 42505
42553 X1=0:Y1=20:GOSUB61000:PRINTSP$;
42555 X1=7:GOSUB61000:PRINT"▯▯  ˎ‾┌‾ˎ▯▯‾ˎ┌/ ‾/ ‾/‾▯┌  "
42560 OPEN2,8,3,"@0:DEF@,SEQ,W"
42562 FOR RE=1 TO M3
42564 R$=""
42568 L=LEN(D$(RE)):IF L<10 THEN D$(RE)=D$(RE)+LEFT$(SP$,10-L)
42570 IF L>10 THEN D$(RE)=LEFT$(D$(RE),10)
42572 XD$=STR$(XD(RE))
42574 IF XD(RE)<10 THENXD$="0"+RIGHT$(XD$,1):GOTO 42580
42576 XD$=RIGHT$(XD$,2)
42580 R$=XD$+RIGHT$(STR$(YD(RE)),2)+D$(RE)
42585 R$=R$+RIGHT$(STR$(XC(RE)),2)
42590 R$=R$+RIGHT$(STR$(YC(RE)),2)
42595 R$=R$+RIGHT$(STR$(LC(RE)),2)
42600 R$=R$+RIGHT$(STR$(TP(RE)),2)
42605 R$=R$+RIGHT$(STR$(ND(RE)),2)
42610 R$=R$+"●●"
42615 PRINT#2,R$
42620 NEXT RE
42625 CLOSE2
42630 OPEN2,8,3,"@0:FOR@,SEQ,W"
42632 FOR RE=1 TO M3
42634 Z$=STR$(RE):IF RE<10 THEN Z$="0"+RIGHT$(Z$,1):GOTO 42638
42636 Z$=RIGHT$(Z$,2)
42638 L=LEN(FO$(RE)):IF L<30 THEN FO$(RE)=FO$(RE)+LEFT$(SP$,30-L)
42640 FO$(RE)=LEFT$(FO$(RE),30)
42645 R$=Z$+FO$(RE)+"●●"
42650 PRINT#2,R$
42655 NEXT RE
42660 CLOSE2
42665 OPEN2,8,3,"@0:PIN@,SEQ,W"
42667 IFII$=""ORII$=SP$+SP$+"  "THENII$="":FORI=1TO82:II$=II$+"#":NEXTI
42669 PRINT#2,II$;
42670 CLOSE2
42680 OPEN2,8,3,"@0:PDF@,SEQ,W"
42685 IF TS$="" THEN TS$="00"
42690 FOR RE=1 TO M3
42695 TS$=TS$+FT$(RE)+NS$(RE)
42700 NEXT RE
42705 PRINT#2,TS$;:CLOSE2
42710 RETURN
43500 REM INSTRUCCIONES
43502 X1=10:Y1=2:PRINTHO$:POKE53280,6
43504 GOSUB61000:PRINT"▯▯┌──────────────────┐"
43506 Y1=Y1+1:GOSUB61000:PRINT"▯▯|‾┘‾▯‾ˎ┌/ ˎ▯▯‾▯‾▯|"
43508 Y1=Y1+1:GOSUB61000:PRINT"▯▯└──────────────────┘"
43510 X1=3:Y1=6
43512 GOSUB61000:PRINT"▯▯|‾/‾ˎ┌/ :"
43514 X1=0:Y1=8:GOSUB61000:PRINT"1) LAS DESCRIPCIONES DEBEN INTRODUCIRSE"
43515 Y1=Y1+1:GOSUB61000:PRINT"   ENTRE DOS CARACTERES & ▯EJ.: &/‾ˎ|‾‾&▯▯"
43517 Y1=Y1+2:GOSUB61000:PRINT"2) LOS CARACTERES ▯▯'#'▯▯,▯▯'$'▯▯,▯▯'%'▯▯,DEFINEN
"
43519 Y1=Y1+1:GOSUB61000:PRINT"   EL TIPO DE CAMPO : ▯#▯ = NUMERICO"
43520 Y1=Y1+1:GOSUB61000:PRINT"   ▯$▯ = ALFANUMERICO, ▯%▯ = DE CALCULO"
43522 Y1=Y1+2:GOSUB61000:PRINT"3) ‾L NUMERO DE CARACTERES ▯#▯,▯$▯,▯%▯ INDICA "
43523 Y1=Y1+1:GOSUB61000:PRINT"   LA LONGITUD DEL CAMPO CONSIDERADO  "
43524 Y1=Y1+1:GOSUB61000:PRINT"   ▯EJ.: $$$$$▯▯= CAMPO ALFANUMERICO"
43525 Y1=Y1+1:GOSUB61000:PRINT"              DE 5 CARACTERES"
43540 Y1=Y1+2:GOSUB61000:PRINT"4) ‾L PUNTO ENTRE LOS CARACTERES ▯▯'#'▯▯ O"
43541 Y1=Y1+1:GOSUB61000:PRINT"   ▯▯'%'▯▯ SEPARA LOS DECIMALES"
43542 Y1=Y1+1:GOSUB61000:PRINT"   ▯EJ.: %%%.%%▯▯= CAMPO DE CALCULO DE 5"
43544 Y1=Y1+1:GOSUB61000:PRINT"   CARACTERES, 2 DE ELLOS DECIMALES"
43545 X1=12:Y1=24:GOSUB61000:PRINT"▯PULSAR UNA TECLA";
43550 GET K$:IF K$="" THEN 43550
43552 FOR I=8TO23:X1=0:Y1=I:GOSUB61000:PRINTSP$;:NEXT I
43560 X1=0:Y1=8:GOSUB61000:PRINT"▯5) LA LONGITUD TOTAL DE LOS CAMPOS, O"
43562 Y1=Y1+1:GOSUB61000:PRINT"   SEA LA SUMA DE TODOS LOS ▯▯'#'▯▯,LOS ▯▯'$'▯▯"
43564 Y1=Y1+1:GOSUB61000:PRINT"   Y ▯▯'%'▯▯ DEBE SER COMO MAXIMO 128"
43568 Y1=Y1+2:GOSUB61000:PRINT"6) LA LONGITUD DE LAS DESCRIPCIONES  "
43570 Y1=Y1+1:GOSUB61000:PRINT"   NO DEBE SER SUPERIOR A 10   "
43575 Y1=Y1+2:GOSUB61000:PRINT"7) ‾N LAS FORMULAS DE CALCULO, EL NUMERO"
43577 Y1=Y1+1:GOSUB61000:PRINT"   DEL CAMPO DEBE ESTAR ENTRE COMILLAS  "
43579 Y1=Y1+1:GOSUB61000:PRINT"   ▯▯ P.E. : 2*('3'+▯▯_('5'))+7.5) ▯▯"
43580 Y1=Y1+2:GOSUB61000:PRINT"8) LAS FUNCIONES UTILIZABLES EN LOS"
43582 Y1=Y1+1:GOSUB61000:PRINT"   CALCULOS SON : ▯▯|▯(N)▯,▯ˎ/|(N)▯,▯‾┌▯(N)▯,"
43584 Y1=Y1+1:GOSUB61000:PRINT"                     ▯▯ˎ/(N)▯,▯▯▯_(N)▯,▯|▯/(N)▯"
43600 GET K$:IF K$="" THEN 43600
43610 RETURN
44000 REM ** AYUDA **
44002 X1=0:Y1=23
44004 GOSUB61000:PRINT"   BARRA ESPACIADORA PARA CONTINUAR    ";
44006 GOSUB 44180
44010 GOSUB61000:PRINT"     ‾|_L+ˎ : INSERCION CARACTER     ";
44012 GOSUB 44180
44014 GOSUB61000:PRINT"     ‾|_L+/ : BORRADO CARACTER    ";
44016 GOSUB 44180
44018 GOSUB61000:PRINT"     ‾|_L+┌ : INTRODUCCION Y CONVAL.    ";
44020 GOSUB 44190
44022 GOSUB61000:PRINT"        ‾|_L+┐ : PIDE LA AYUDA         ";
44024 GOSUB 44180
44026 GOSUB61000:PRINT"     _‾|/_/ : LLEVA CURSOR AL PRINCIPIO    ";
44028 GOSUB 44180
```

```
44030 GOSUB61000:PRINT" ┤┴+♠ : DESPLAZA CURSOR A LA IZQUIERDA ";
44032 GOSUB 44180
44034 GOSUB61000:PRINT"  ┤┴+● : DESPLAZA CURSOR A LA DERECHA  ";
44036 GOSUB 44180
44038 GOSUB61000:PRINT"  ┤┴+o : DESPLAZA CURSOR ARRIBA       ";
44040 GOSUB 44180
44042 GOSUB61000:PRINT"  ┤┴+◆ : DESPLAZA CURSOR ABAJO  ";
44044 GOSUB 44180
44046 GOSUB61000:PRINTLEFT$(SP$,39);
44048 RETURN
44180 REM
44182 GET K$:IF K$<>CHR$(32) THEN 44182
44190 RETURN
44500 REM LECTURA DATOS
44505 PRINTHO$:X1=12:Y1=6:GOSUB61000:PRINT"▓♠ | ‾ ⁄ ‾ ╲ ┌ ⁄"
44506 X1=X1-2:Y1=Y1+2:GOSUB61000:PRINT"▊▔┤ ⁄▃♠ ▔▔╲⁄╲▔╲┌⁄●"
44507 X1=5:Y1=12:GOSUB61000
44510 PRINT"♠⁄┤┌▔⁀▁ ▔⁄ L♠ ⁄⁄╲▔●▔⁄ ▔╲●┤▔"
44512 X1=1:Y1=Y1+2:GOSUB61000:PRINT"▊┌L ●┌┬┤▔ ▁●▔✕♠┬ ♠ L♠● ▔▔╲⁄╲▔╲┌⁄●"
44514 X1=11:Y1=20:GOSUB61000:PRINT"▓┐⁄L●♠▁ ⁄♠ |▔┤L♠":POKE 198,0:WAIT 198,1
44520 X1=0:Y1=20:GOSUB61000:PRINTSP$;
44525 OPEN1,8,15,"I/O":GOSUB63000:IF AD<>0 THEN RETURN
44530 OPEN2,8,3,"0:DEF@,SEQ,R"
44535 INPUT#1,AD,X$,CD,DD:IF AD=62 THEN GOSUB63050:CLOSE1:RETURN
44540 FOR RE=1 TO M3
44545 INPUT#2,R$
44550 XD(RE)=VAL(MID$(R$,1,2))
44555 YD(RE)=VAL(MID$(R$,3,2))
44560 D$(RE)=MID$(R$,5,10)
44565 XC(RE)=VAL(MID$(R$,15,2))
44570 YC(RE)=VAL(MID$(R$,17,2))
44575 LC(RE)=VAL(MID$(R$,19,2))
44580 TP(RE)=VAL(MID$(R$,21,2))
44585 ND(RE)=VAL(MID$(R$,23,2))
44590 NEXT RE:CLOSE2
44595 OPEN2,8,3,"0:FOR@,SEQ,R"
44600 INPUT#1,AD,X$,CD,DD:IF AD=62 THEN GOSUB63050:CLOSE1:RETURN
44605 FOR RE=1 TO M3
44610 INPUT#2,R$
44615 FO$(RE)=MID$(R$,3,30)
44620 NEXT RE:CLOSE2
44700 CLOSE1:RETURN
45000 REM
45001 REM DEFINICIONES PARA IMPRESION
45002 REM
45025 IF W5=0 THEN W5=2:GOSUB 44500:IF AD<>0 THEN W5=0:RETURN
45027 POKE198,0:I$=""
45030 PRINTHO$:POKE646,1:X1=0:Y1=0:GOSUB61000:PRINTH1$:X1=0:Y1=6:GOSUB61000
45032 PRINT"╲NTRODUCIR LA CABECERA  (MAX. 80 CAR.) ╓?":A=0
45034 GET H$:IF H$="" THEN 45034
45035 A1=ASC(H$):IF A1=13 THEN 45060
45036 IF A1=20 THEN PRINTH$;:A=A-1:I$=LEFT$(I$,A):GOTO 45034
45037 IF(A1<65ORA1>90)AND(A1<193ORA1>218)AND(A1<48ORA1>57)ANDA1<>32THEN45034
45038 I$=I$+H$:PRINTH$;
45040 A=A+1:IF A<80 THEN 45034
45042 IF A=80 THEN XP=0:GOTO 45090
45060 IF LEN(I$)<=0 THEN 45000
45062 X1=0:Y1=13:GOSUB61000:PRINT"♠NTRODUCIR COLUMNA PRINCIPIO IMPRESION"
45070 X1=10:Y1=18:GOSUB61000:INPUT "╓";XP:IFXP<0ORXP>80-ATHEN 45030
45090 IFXP<10THENA$="0"+RIGHT$(STR$(XP),1):GOTO 45100
45095 A$=RIGHT$(STR$(XP),2)
45100 II$=A$+I$+LEFT$(S1$,80-LEN(I$))
45110 REM
45120 REM TIPO DE IMPRESION
45130 REM
45140 PRINTHO$:POKE646,1:X1=0:Y1=0:GOSUB61000:PRINTH1$:X1=5:Y1=8:GOSUB61000
45142 PRINT"▔LEGIR EL TIPO DE IMPRESION:"
45145 X1=11:Y1=12:GOSUB61000:PRINT"╓1)▐ HORIZONTAL"
45150 Y1=Y1+2:GOSUB61000:PRINT"╓2)▐ VERTICAL   "
45170 GET A$:IF A$<>"1" AND A$<>"2" THEN 45170
45180 TS$="0"+A$
45190 PRINTHO$:X1=0:Y1=0:GOSUB61000:PRINTH1$
45200 X1=2:Y1=5:GOSUB61000:PRINT"╲N LA SIGUIENTE MASCARA DEBERAN";
45202 PRINT"▌        SELECCIONARSE LOS CAMPOS A IMPRIMIR,"
45210 X1=2;Y1=9:GOSUB61000:PRINT"INSERTANDO LA LETRA '▓┬▐' SI EL CAMPO"
45212 PRINT"▌  (NUMERICO O DE CALCULO) DEBE ▌          TOTALIZARSE."
45220 Y1=17:GOSUB61000:PRINT"▓EJ.: 2-T1-15-T6▐= IMPRESION DE LOS"
45221 PRINT"  CAMPOS ╓2▐,╓1▐,╓15▐ Y ╓6▐ CON TOTALIZACION"
45222 PRINT"  DE LOS CAMPOS ╓1▐ Y ╓6▐"
45230 X1=11:Y1=23:GOSUB61000:PRINT"▓PULSAR UNA TECLA▐";
45240 POKE198,0:WAIT198,1
45250 PRINTHO$:GOSUB41020:FORI=1TOM3:IFLEFT$(D$(I),3)="   "THEN 45270
45260 X1=XD(I):Y1=YD(I):GOSUB61000:PRINTD$(I)
45270 NEXT I
45280 FORI=1TOM3:IFTP(I)=0THEN45300
45290 X1=XC(I):Y1=YC(I):GOSUB61000:PRINTSTR$(I)
45300 NEXT I
45310 X1=0:Y1=23:GOSUB61000:PRINT"INDICA CAMPOS A IMPRIMIR (P.E. 2-T3-1)";:POKE1
98,0
45320 Y1=Y1+1:GOSUB61000:INPUTB$:LL=LEN(B$)
45330 A$="":C=0:FORI=1TOLL
45340 C$=MID$(B$,I,1):IFC$="-"THENGOSUB45370:GOTO45360
45350 A$=A$+C$
45360 NEXT I:I=I-1:GOSUB45370:RETURN
45370 REM
45380 C=C+1
45390 L=LEN(A$):IFLEFT$(A$,1)="T"THENFT$(C)="01":L=L-1:A$=MID$(A$,2,L):GOTO45400
45395 FT$(C)="00"
45400 V=VAL(A$):IFV=0THEN45470
45410 IFTP(V)=0THEN45470
45420 IFTP(V)=3 THEN FT$(C)="00"
45430 IFL>2ORL<1THEN 45470
45440 IFL=1THENNS$(C)="0"+A$:GOTO 45450
45445 NS$(C)=A$
45450 IF I=LL THEN 45560
45460 A$="":RETURN
45470 REM ERROR
45480 PRINTHO$:X1=15:Y1=11:GOSUB61000:PRINT"▔ ▁ ▁ ┌ ▁"
45490 FORT=1TO1200:NEXTT:FORI=1TOM3:FT$(I)="":NS$(I)="":NEXTI
45500 PRINTHO$:X1=10:Y1=10:GOSUB61000:PRINT"▓┐ 1 ▆▐ CONTINUO"
45510 Y1=Y1+2:GOSUB61000:PRINT"▓┐ 2 ▆▐ VUELVO AL MENU"
45520 X1=13:Y1=17:GOSUB61000:PRINT"┤UAL ?"
45530 GET A$:IF A$<>"1" AND A$<>"2" THEN45530
45540 IF A$="2" THEN RETURN
45550 GOTO 45250
45560 FORI=C+1 TO M3
45570 FT$(I)="00":NS$(I)="00":NEXT
45580 REM
45590 IFTS$="02" THEN 45640
45600 B=1:FORI=1TOM3:A=VAL(NS$(I)):IFA=0THEN45620
45610 B=B+LC(A)+1
45620 NEXT I
45630 IF B>80THEN 45470
45640 RETURN
46000 REM
46001 REM MENU
46002 PRINTHO$:POKE646,1:X1=0:Y1=1:GOSUB61000:PRINTH1$
46005 X1=12:Y1=5:GOSUB61000:PRINT"╲▔⁄⁄ ▔▔╲⁄╲▔╲┌⁄●"
46007 X1=12:Y1=6:GOSUB61000:PRINT"▓──────────────▐"
46010 X1=4:Y1=11:GOSUB61000:PRINT"▓┐ 1 ▆▐ ┤REACION MASCARA"
46012 Y1=13:GOSUB61000:PRINT"▓┐ 2 ▆▐ ▔EFINICION FORMATOS IMPRESION"
46014 Y1=15:GOSUB61000:PRINT"▓┐ 3 ▆▐ |RABACION FICHEROS DEFINICIONES"
46016 Y1=17:GOSUB61000:PRINT"▓┐ 4 ▆▐ ▁IN DE TAREA "
46050 POKE 198,0
46100 X1=12:Y1=21:GOSUB61000:INPUT"▐┤UAL : █";A$
46110 OPZ=VAL(A$):IF OPZ<1 OR OPZ>4 THEN 46100
46115 IF OPZ=4 THEN PRINTHO$:POKE 646,14:END
46120 RETURN
55000 REM RESTABLECIMIENTO
55005 L1=1024+(40*Y2)+X2:C1=PEEK(L1)
55009 IF C1=160 THEN C1=32:GOTO 55015
55010 IF C1>128 THEN C1=C1-128
55015 POKE L1,C1:RETURN
60000 REM PRINT AT
60005 POKE 211,X:POKE 214,Y
60010 SYS58732:RETURN
61000 REM PRINT AT II
61005 POKE 211,X1:POKE 214,Y1
61010 SYS58732:RETURN
62000 REM CONVERSIONES CARACTERES
62005 IF CH=160THEN CH=32:RETURN
62010 IF CH>128 THEN CH=CH-128:RETURN
62015 IF CH<32 THEN CH=CH+64:RETURN
62020 RETURN
63000 REM CONTROLES DISCO
63005 ER=0:INPUT#1,AD,X$,CD,DD
63011 IF AD=0 THEN RETURN
63012 IF AD=21 THEN X$="     ERROR EN UNIDAD DISCO     ":ER=1
```

```
63020 IF AD=72 THEN X$="ESPACIO UTIL EN DISCO AGOTADO":ER=2
63022 IF AD=74 THEN X$=" UNIDAD DISCO NO EN LINEA   ":ER=3
63041 X1=0:GOSUB61000:PRINTSF$;
63042 X1=5:GOSUB61000:PRINT"▀";X$:POKE198,0:WAIT198,1
63045 RETURN
63050 X$="   NO HAY EL FICHERO DATOS    ":ER=4:GOTO 63041
```

Main

```
5 REM ***********************
6 REM *                     *
7 REM * BASE DE DATOS C-64 *
8 REM *                     *
9 REM ***********************
13 PRINT"♥":PRINTCHR$(14);:M3=20:MR=100
15 DIM HLN(M3),HND(M3),IS(MR),NS(M3),FT(M3),Z8(M3)
20 DIM B$(M3),TP(M3),LC(M3),ND(M3),CX(M3),CY(M3),DX(M3),DY(M3),HTP(M3)
21 DIM TF(15),D$(M3),H$(12),F$(M3),DA$(MR)
24 BL$=" ":PD$=".":AS$="$":NS$="#":SU$="▌":GIU$="▐":HO$="♥":SL$="/"
25 POKE 650,128
26 PRINTHO$
27 REM *** LECTURAS DE SERVICIO ***
30 RESTORE
35 FOR I=1 TO 15:READ TF(I):NEXTI
40 FOR I=1 TO 12:READ H$(I):NEXTI
41 READ H1$:READ PROG$
42 READ VN:FORI=1TOVN:READ VO$(I):NEXT
43 VO$(I)="_IN DE TAREA"
45 SP$="":FOR I=1 TO 40:SP$=SP$+CHR$(32):ZE$=ZE$+"00":NEXT
50 Q$="":F$="-":FOR I=1 TO 38:Q$=Q$+F$:NEXT I
60 FORI=1TO5:READOP$(I):NEXT
70 FORI=1TO6:READFZ$(I):NEXT
100 REM
219 REM ------------------------
220 REM ------ MAIN ------------
221 REM ------------------------
230 GOSUB 1300:REM MENU
235 PRINTHO$
240 IFOPZ=0THENFORI=1TO1000:NEXT:PRINTHO$:PRINTCHR$(142):POKE646,14:END
250 ON OPZ GOSUB 4500,5500,14000,12000,16000
290 GOTO 230
600 REM ESPERAR
610 PRINTHO$:X=13:Y=12:GOSUB11000:PRINT"▀▀ ♦ ┐ ¯ _ ♠ _▀▀":RETURN
1300 REM ------------------------
1310 REM ------- MENU -------------
1320 REM ------------------------
1322 PRINTHO$:POKE198,0:AD=0:X$="":CD=0:DD=0
1335 X=1:Y=0:GOSUB11000:PRINTH1$
1340 L=LEN(PROG$):LL=(40-L)/2
1345 X=LL:Y=3:GOSUB11000:PRINT"▀"PROG$
1347 X=LL-1:Y=4:GOSUB11000:FORI=1TOL+2:PRINT"▀";:NEXT
1350 FORI=1TOVN+1:X=7:A=I:IFI=VN+1THENA=0
1360 Y=5+I*2:GOSUB11000:PRINT"▀▀";A;"▌ ▀▀  ";VO$(I);:NEXT
1370 X=26:Y=23:GOSUB11000:PRINT"▀UAL? (   )";
1380 GETI$:OPZ=VAL(I$):IFOPZ<=VNANDOPZ>0ORI$="0"THEN1400
1390 GOTO1380
1400 X=33:Y=23:GOSUB11000:PRINT"▀";OPZ;"▌ "
1410 FORI=1TO1000:NEXT:RETURN
2000 REM
2005 REM *** PRESENTA MASCARA ***
2010 NUC=M3:PRINTHO$:X=1:Y=0:GOSUB11000:PRINTH1$:POKE 646,7
2011 X=0:Y=1:GOSUB11000:PRINT"┌"+Q$+"┐";
2015 FORY=2TO21:GOSUB11000:PRINT"|";+LEFT$(SP$,38)+"|";:NEXT Y
2017 X=0:Y=22:GOSUB11000:PRINT"└"+Q$+"┘";:POKE 646,1
2020 FOR J=1 TO NUC:IF D$(J)=LEFT$(SP$,10) THEN 2025
2022 X=DX(J):Y=DY(J):GOSUB11000:PRINT"▀";D$(J)
2025 NEXT J
2030 FOR J=1 TO NUC
2032 A$=""
2035 IF TP(J)=0 THEN 2130
2050 IF TP(J)<>5 AND TP(J)<>4 THEN 2060
2054 X=CX(J):Y=CY(J):GOSUB11000:PRINT"▶";B$(J)
2056 GOTO 2130
2060 N1=LC(J):N2=ND(J):IF TP(J)=1 THEN C$=NS$:GOTO2070
2062 IF TP(J)=2 THEN C$="%":GOTO2070
2064 C$=AS$
2065 IF N1=1 THEN 2075
2070 FOR I=1 TO N1-N2-1:A$=A$+C$:NEXTI
2075 IF N2=0 THEN A$=A$+C$:GOTO 2095
2080 A$=A$+PD$
2090 FOR I=1 TO N2:A$=A$+"0":NEXTI
2095 X=CX(J):Y=CY(J):GOSUB 11000
2096 PRINT "▀";A$;"▶"
2097 B$(J)=""
2098 FOR K=1 TO N1-N2-1
2100 B$(J)=B$(J)+BL$
2105 NEXT K
2106 IF N2=0 THEN B$(J)=B$(J)+BL$:GOTO 2130
2110 B$(J)=B$(J)+PD$
2120 FORK=1TON2:B$(J)=B$(J)+BL$:NEXTK
2130 NEXT J
2150 RETURN
3000 REM
3010 REM ** DATA ENTRY **
3030 J=1:NUC=M3:POKE198,0
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOTO 3240
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 REM CARACTER
3105 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 FS=0:J=NUC:GOTO 3040
3150 IF J<NUC THEN J=J+1:GOTO 3040
3160 FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB11000:GOTO 3060
3200 FS=0:GOTO 3040
3210 REM
3220 IF FS>8 THEN FS=0:GOTO 3060
3230 IF FS=8 THEN FS=0:GOTO 3100
3235 RETURN
3240 REM
3250 REM  ** LECTURA DE 1 CAMPO **
3280 FS=0
3285 IF TP(NC)=0 THEN 3070
3290 IF TP(NC)=4 THEN 3070
3300 IF TP(NC)<>2 THEN 3310
3302 GOTO 8500
3304 GOTO 3070
3310 IF MID$(B$(NC),CC,1)=PD$ THENX=X+S:CC=CC+S
3320 GOSUB 11000
3330 GET A$:IF A$<>"" THEN 3338
3332 GOSUB11000:PRINT"▀";:GOSUB11000:PRINT"▶";MID$(B$(NC),CC,1);
3336 GOTO 3330
3338 IFASC(A$)=19ORASC(A$)=20ORASC(A$)=147ORASC(A$)=148 ORASC(A$)=141 THEN 3330
3340 PRINT"▌▀";A$;
3350 CA=ASC(A$)
3360 FOR I=1 TO 15
3370 IF TF(I)=CA THEN FS=I:I=15:GOSUB3630:GOTO 3070
3375 IF FS=9 THEN 3070
3380 NEXTI
3390 IF TP(NC)<>1 THEN 3450
3400 IF CA=45 OR CA=46 OR(CA>=48ANDCA<=57)THEN3420
3410 GOTO 3330
3420 IF CA<>46 THEN3450
3430 GOSUB 4270
3440 IF SWP=1 THEN 3410
3450 S=1:GOSUB 3520
3460 IF CC=LC(NC) THEN 3070
3470 CC=CC+1:X=X+1:GOTO 3310
3520 REM
3521 REM ** TRASFERENCIA AL BUFFER **
3540 REM
3570 IF CC=1 THEN SN$="":GOTO 3590
3580 SN$=LEFT$(B$(NC),CC-1)
3590 IF CC=LC(NC) THEN DE$="":GOTO3610
3600 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3610 B$(NC)=SN$+A$+DE$
3620 RETURN
3625 REM
3630 REM ** COMANDOS **
```

```
3640 REM
3680 IF FS>5 THEN 3710
3700 RETURN
3710 IF FS=9 AND OPZ<>2 THEN GOSUB 13000:RETURN
3720 IF FS=8 THEN GOSUB 4080:RETURN
3725 IF FS<8 THEN RETURN
3730 ON FS-9 GOSUB 3740,3910
3735 RETURN
3740 REM
3745 REM ** INSERCION **
3750 REM
3770 IF CC=LC(NC)THEN RETURN
3780 IF CC=1 THEN SN$="":GOTO3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DE$=RIGHT$(B$(NC),N)
3820 C$=BL$
3830 A$=SN$+C$+DE$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 REM
3890 PRINT B$(NC):S=-1
3900 IF RIGHT$(B$(NC),1)=PD$ THEN 3780
3905 RETURN
3909 REM
3910 REM ***** BORRADO *****
3920 REM
3930 IF TP(NC)<>1 THEN 3950
3940 IF MID$(B$(NC),2,1)=PD$ AND CC=1 THEN RETURN
3950 IF CC<1 THEN RETURN
3960 IF CC=LC(NC) THEN DE$="":GOTO 3990
3970 N=LC(NC)-CC:DE$=RIGHT$(B$(NC),N)
3980 SN$=LEFT$(B$(NC),CC-1)
3990 IF TP(NC)=1 OR TP(NC)=2 THEN C$=NS$:GOTO 4010
4000 C$=AS$
4010 B$(NC)=SN$+DE$+C$
4020 XX=X:YY=Y
4030 X=CX(NC):Y=CY(NC):GOSUB 11000
4040 X=XX:Y=YY
4050 PRINT B$(NC):S=1:RETURN
4080 REM ******* ALINEADOS *******
4084 IF LC(NC)=1 THEN RETURN
4085 N1=LC(NC)-CC+1
4087 IF CC=1 THEN B$(NC)="":GOTO4100
4090 B$(NC)=LEFT$(B$(NC),CC-1)
4100 IF TP(NC)=1 OR TP(NC)=2 THEN 4150
4110 FOR I=1 TO N1
4120 B$(NC)=B$(NC)+BL$
4130 NEXT I
4140 GOTO  4190
4150 FOR I=1 TO N1
4160 B$(NC)=CHR$(32)+B$(NC)
4170 NEXT I
4180 IF RIGHT$(B$(NC),1)=PD$ THEN B$(NC)=BL$+LEFT$(B$(NC),LC(NC)-1)
4190 X=CX(NC):Y=CY(NC):GOSUB 11000
4210 PRINT B$(NC):RETURN
4270 REM *** CTRL FUNTO ***
4290 SWP=0
4300 IF CC=1 OR CC=LC(NC) THEN SWP=1:RETURN
4310 FOR I=1 TO LC(NC)
4320 IF MID$(B$(NC),I,1)=BL$ THEN I=LC(NC):SWP=1:RETURN
4330 NEXT I:RETURN
4339 REM
4340 REM ******************************
4341 REM
4500 REM
4501 REM INTRODUCCION DATOS
4502 REM
4530 GOSUB 7000:IF AD<>0 THEN RETURN
4535 PRINTHO$:X=7:Y=10:GOSUB11000:PRINT"■ 1 ■ EL FICHERO DATOS EXISTE"
4540 Y=Y+2:GOSUB11000:PRINT"■ 2 ■ CREACION FICHERO DATOS"
4545 X=13:Y=19:GOSUB11000:PRINT"-UAL  ?":X=20:GOSUB11000:POKE 204,0
4550 GET R$:IF R$<>"1" ANDR$<>"2"THEN4550
4570 POKE204,1:GOSUB11000:PRINTR$:FORT=1TO400:NEXTT
4580 IF R$="1" THEN GOSUB 5200:IF AD=62 THEN RETURN
4585 GOTO 4600
4590 MX=0:GOTO 4650
4600 MX=VAL(DA$(0))
4650 REM ! INICIO BUCLE INTRODUCCION !
4660 FOR KJ=MX+1 TO MR:H=KJ
4670 GOSUB2000:X=2:Y=23:GOSUB11000:PRINT"■EGISTRO N.";H;"■"
4672 X=26:GOSUB11000:PRINT"■"
4675 GOSUB 3000
4680 IF FS=6 THEN 4670
4682 IF FS=7 THEN RETURN
4690 DA$(KJ)="*"
4700 FOR I=1 TO M3:IF TP(I)=0 THEN 4720
4710 DA$(KJ)=DA$(KJ)+B$(I)
4720 NEXT I
4725 IF KJ<10 THEN DA$(0)="00"+RIGHT$(STR$(KJ),1):GOTO 4740
4730 IF KJ<100 THEN DA$(0)="0"+RIGHT$(STR$(KJ),2):GOTO 4740
4735 DA$(0)=RIGHT$(STR$(KJ),3)
4740 PRINTHO$:X=8:Y=11:GOSUB11000:PRINT"■TRA INTRODUCCION (■/■) ?":X=33
4742 GOSUB11000:POKE204,0
4745 GET A$:IF A$<>"S" AND A$<>"N" THEN 4745
4747 POKE204,1
4750 IF A$="N" OR A$="/" THEN 4780
4770 NEXT KJ:KJ=KJ-1
4780 GOSUB 5000:RETURN
5000 REM
5001 REM GRABACION DATOS
5002 CLOSE15:CLOSE2:CLOSE1
5005 PRINTHO$:X=3:Y=9:GOSUB11000:PRINT"■OMBRE DEL FICHERO "
5006 Y=Y+1:GOSUB11000:INPUT"(MAX. 6 CARACTERES) ■";NF$
5007 IF LEN(NF$)<=0 OR LEN(NF$)>6 THEN 5005
5008 OPEN1,8,15,"I/O":INPUT#1,AD,X$,CD,DD
5010 IF AD<>0 THEN GOSUB 9900:CLOSE1:GOTO 5005
5015 OPEN2,8,3,"@0:"+NF$+",SEQ,W"
5017 PRINT#2,DA$(0)
5020 FOR I=1 TO VAL(DA$(0))
5025 PRINT#2,DA$(I);
5027 NEXT I:CLOSE2:CLOSE1
5050 Q6=2:RETURN
5055 REM -------------------
5200 REM
5201 REM LECTURA DATOS
5202 CLOSE1:CLOSE2:CLOSE15
5203 IF Q6=2 THEN RETURN
5205 PRINTHO$:X=7:Y=8:GOSUB11000:PRINT"■INTRODUCIR EL DISCO DATOS"
5206 X=X+4:Y=Y+2:GOSUB11000:PRINT"Y PULSAR UNA TECLA"
5207 POKE 198,0:WAIT 198,1
5210 POKE 198,0
5215 X=3:Y=13:GOSUB11000:PRINT "■OMBRE DEL FICHERO "
5217 Y=Y+1:GOSUB11000:INPUT"(MAX. 6 CARACTERES) ■";NF$
5218 IF LEN(NF$)<=0 OR LEN(NF$)>6 THEN 5205
5220 OPEN1,8,15,"I/O":INPUT#1,AD,X$,CD,DD
5225 IF AD<>0 THEN GOSUB 9900:CLOSE1:RETURN
5230 OPEN2,8,3,"0:"+NF$+",SEQ,R":INPUT#1,AD,X$,CD,DD
5232 IF AD=62 THEN GOSUB 9908:CLOSE1:CLOSE2:RETURN
5235 INPUT#2,DA$(0)
5237 LR=0:FOR J1=1 TO 20:LR=LR+LC(J1):NEXT J1:LR=LR+1
5240 FOR I=1 TO VAL(DA$(0))
5242 DA$(I)=""
5245 FOR J2=1 TO LR:GET#2,A1$:DA$(I)=DA$(I)+A1$:NEXT J2
5247 NEXT I
5250 CLOSE1:Q6=2:RETURN
5500 REM
5501 REM ORDENADO
5502 REM
5540 GOSUB 7000:IF AD<>0 THEN RETURN
5545 GOSUB5200:IF AD<>0 THEN RETURN
5550 FOR I=1TOM3:HTP(I)=TP(I):HLC(I)=LC(I):HND(I)=ND(I)
5570 IF TP(I)<>0 THEN TP(I)=3:LC(I)=1:ND(I)=0
5580 NEXT I
5590 GOSUB 2000:X=1:Y=23:GOSUB11000:PRINT"■'*' EN EL CAMPO CLAVE DEL ORDENADO";
5592 FK=2:GOSUB 3000:FK=0
5600 IF FS=6 THEN 5590
5605 FORI=1TOM3:TP(I)=HTP(I):ND(I)=HND(I):LC(I)=HLC(I):NEXT
5608 IF FS=7 THEN RETURN
5610 FORI=1TOM3:IFB$(I)="*" THEN K4=I:GOTO5630
5620 NEXTI:PRINTHO$:X=4:Y=10:GOSUB11000:PRINT"EL CAMPO CLAVE NO SE HA INDICADO"
5625 X=12:Y=13:GOSUB11000:PRINT"VUELVO AL MENU":FORI=1TO2500:NEXTI:RETURN
5630 IFK4=1THENP1=2:P2=LC(1):GOTO5670
5640 P1=1:P2=0:FORI=1TOK4-1
5650 P1=P1+LC(I):NEXT I
5660 P1=P1+1:P2=LC(K4)
5670 GOSUB 600:GOSUB 6000:W6=2:P8=P1:P9=P2
```

```
5680 PRINTHO$:X=11:Y=11:GOSUB11000:PRINT"Γ–¯/♠┬ ¯–¯┤ ♠┬":FORI=1TO1000:NEXTI
5690 RETURN
6000 REM
6001 REM          SORT
6002 REM
6010 N=VAL(DA$(0))
6020 FORI=1TON:IS(I)=I:NEXTI
6040 I1=1:J1=N
6050 I=I1:J=J1:S=-1
6060 GOSUB 6220
6065 IF TP(K4)<3 THEN IF VAL(A1$)<=VAL(B1$) THEN 6100
6067 IF TP(K4)<3 THEN 6080
6070 IF A1$<=B1$ THEN 6100
6080 V=IS(I):IS(I)=IS(J)
6090 IS(J)=V:S=SGN(-S)
6100 IFS=1THEN I=I+1:GOTO6120
6110 J=J-1
6120 IF I<J THEN 6060
6130 IFI+1>J1 THEN 6160
6140 P=P+1
6150 S9(P,1)=I+1:S9(P,2)=J1
6160 J1=I-1
6170 IF I1<J1 THEN 6050
6180 IF P=0 THEN 6210
6190 I1=S9(P,1):J1=S9(P,2):P=P-1
6200 GOTO 6050
6210 RETURN:REM FIN
6220 A1$=MID$(DA$(IS(I)),P1,P2)
6230 B1$=MID$(DA$(IS(J)),P1,P2)
6240 RETURN
6245 REM ----------------------------
7000 REM
7001 REM *** LECTURA DATOS MASCARA ***
7002 IF Q8=2 THEN RETURN
7006 PRINTHO$:X=11:Y=8:GOSUB11000:PRINT"■ ♠ | ¯ / ¯ ╲ Γ / "
7007 X=X-2:Y=10:GOSUB11000:PRINT"▌L¯┤ ∠♠ ¯¯╲∕╲¯╲Γ∕¯●"
7008 X=2:Y=Y+3:GOSUB11000:PRINT"INTRODUCIR EL DISCO QUE CONTIENE LOS"
7009 Y=Y+2:GOSUB11000:PRINT"FICHEROS DE DEFINICION DE MASCARAS"
7010 X=12:Y=20:GOSUB11000:PRINT"■Π∠●♠– ∕♠ |¯¯L♠":POKE198,0:WAIT198,1
7012 OPEN1,8,15,"I/O":INPUT#1,AD,X$,CD,DD
7014 IF AD<>0 AND AD<>62 THEN GOSUB9900:RETURN
7017 OPEN2,8,3,"0:DEF@,SEQ,R":INPUT#1,AD,X$,CD,DD
7018 IF AD=62 THEN GOSUB 9908:RETURN
7020 FOR I=1 TO M3
7022 INPUT#2,A$
7024 DX(I)=VAL(LEFT$(A$,2)):DY(I)=VAL(MID$(A$,3,2)):D$(I)=MID$(A$,5,10)
7026 CX(I)=VAL(MID$(A$,15,2)):CY(I)=VAL(MID$(A$,17,2))
7028 LC(I)=VAL(MID$(A$,19,2)):TP(I)=VAL(MID$(A$,21,2)):ND(I)=VAL(MID$(A$,23,2))
7030 NEXT I:CLOSE2
7032 OPEN2,8,3,"0:FOR@,SEQ,R":INPUT#1,AD,X$,CD,DD
7034 IF AD=62 THEN GOSUB 9908:RETURN
7036 FOR I=1 TO M3
7038 INPUT#2,A$:F$(I)=MID$(A$,3,30)
7040 NEXT I:CLOSE2
7042 OPEN2,8,3,"0:PIN@,SEQ,R"
7044 A$="":FOR RE=1 TO 82:GET#2,A1$:A$=A$+A1$:NEXT RE:CLOSE2
7046 XP=VAL(LEFT$(A$,2)):II$=MID$(A$,3,80)
7050 OPEN2,8,3,"0:PDF@,SEQ,R"
7051 REM
7052 A$="":FOR RE=1 TO 82:GET#2,A1$:A$=A$+A1$:NEXT RE:CLOSE2
7053 REM
7054 FI=VAL(LEFT$(A$,2))
7056 C1=3:FOR I=1 TO M3
7058 FT(I)=VAL(MID$(A$,C1,2)):NS(I)=VAL(MID$(A$,C1+2,2)):C1=C1+4
7060 NEXT I:A$=""
7062 L=LEN(II$):IF L=0 THEN 7110
7064 FOR I=LEN(II$) TO 1 STEP-1
7068 IF MID$(II$,I,1)<>" "THEN 7100
7070 L=L-1:NEXT I
7100 II$=LEFT$(II$,L)
7110 CLOSE1:Q8=2:RETURN
8500 REM  PREPARA FORMULA
8520 A$=F$(NC)
8530 L=LEN(A$):FORI=1TOL
8540 IFMID$(A$,I,1)=CHR$(39)THEN8560
8550 NEXTI:GOTO8620
8560 A1$=MID$(A$,I+1,1)
8562 IFMID$(A$,I+2,1)<>CHR$(39)THENA1$=A1$+MID$(A$,I+2,1):P=4:GOTO8570
8565 P=3
```

```
8570 A=VAL(A1$):L1=LEN(B$(A)):P1=1:FORJJ=1TOL1
8580 IFMID$(B$(A),JJ,1)=" "THENP1=P1+1
8590 NEXTJJ:A2$=MID$(B$(A),P1,40):IFVAL(A2$)=0THENA2$="0"
8600 A$=MID$(A$,1,I-1)+A2$+MID$(A$,I+P,L)
8610 GOTO8530
8620 L2=LEN(A$):C$="":FORI=1TOL2:CH$=MID$(A$,I,1)
8621 IF CH$<>" "THEN C$=C$+CH$
8622 NEXT I:A$=C$:C$=""
8624 GOTO 9130
8625 IFER<>0THENB$(NC)="$":L=1:GOSUB8700:NC=0:GOTO 8695
8630 L=LEN(A$):PP=1
8640 FORK1=1TOL:IFMID$(A$,K1,1)="."THEN8657
8650 NEXTK1:A1$=A$:A2$=".":GOTO8660
8657 A1$=LEFT$(A$,K1-1):IFND(NC)<>0THENA2$=MID$(A$,K1,ND(NC)+1)
8660 IFND(NC)=0THENA2$="":PP=0:GOTO8670
8665 IF(LEN(A2$)-1)<ND(NC)THENA2$=A2$+MID$(ZE$,1,ND(NC)-LEN(A2$)+1)
8670 N=LC(NC)-ND(NC)-PP:IFLEN(A1$)>NTHENB$(NC)="$":L=1:GOSUB8700:NC=0:GOTO8695
8680 B$(NC)=A1$+A2$:L=LEN(B$(NC))
8690 GOSUB8700
8695 GOTO 3304
8700 REM ** A LA DERECHA **
8705 A$=MID$(SP$,1,LC(NC)-L)+B$(NC):X=CX(NC):Y=CY(NC):GOSUB11000:PRINTA$
8710 B$(NC)=MID$(SP$,1,LC(NC)-L)+B$(NC):RETURN
9000 REM
9005 ER=0:L=LEN(CA$):IFL<3THEN9122
9010 FORI=2TOL-1:CH$=MID$(CA$,I,1):FORJJ=1TO5:IFOP$(JJ)=CH$THEN9020
9015 NEXTJJ,I:ER=1:GOTO9122
9020 CH$=MID$(CA$,I+1,1):IFCH$="*"ORCH$="↑"ORCH$="/"THENER=2:GOTO9122
9025 N1=VAL(MID$(CA$,1,I-1)):N2=VAL(MID$(CA$,I+1,40))
9026 ONJJGOTO9035,9040,9050,9055,9060
9027 CA$=STR$(N1):IFN1>=0THENCA$="+"+MID$(CA$,2,80)
9030 GOTO9122
9035 N1=N1↑N2:GOTO9027
9040 IFN2=0THENER=3:GOTO9027
9045 N1=N1/N2:GOTO9027
9050 N1=N1*N2:GOTO9027
9055 N1=N1-N2:GOTO9027
9060 N1=N1+N2:GOTO9027
9065 REM
9070 ER=0:LO=LEN(FO$):IFLO>=3THEN9075
9072 IFEG=1THEN9192
9074 GOTO9162
9075 FOROP=1TO5:PO=0:P1=0:P2=0:FORIO=2TOLO-1:CH$=MID$(FO$,IO,1):CH=ASC(CH$)
9076 IFCH$=OP$(OP)ANDPO=0THENPO=IO:GOTO9095
9080 O=0:IF(CH=94ORCH<48)ANDCH<>46THENO=1
9085 IFO=1ANDPO=0THENP1=IO
9090 IFO=1ANDPO>0ANDIO-PO>1THENP2=IO:GOTO9105
9095 NEXTIO:IFPO>0THEN9105
9100 NEXTOP:IFEG=1THEN9192
9102 GOTO9162
9105 IFP1=0THENP1=1:GOTO9115
9110 CH$=MID$(FO$,P1,1):IFCH$<>"+"ANDCH$<>"-"THENP1=P1+1
9115 P2=P2-1:IFP2<0THENP2=LO
9120 CA$=MID$(FO$,P1,P2-P1+1):GOTO9000
9122 IFER=0THEN9125
9123 IFEG=1THEN9192
9124 GOTO9162
9125 GOTO9280
9127 FO$=MID$(FO$,1,P1-1)+CA$+MID$(FO$,P2+1,40):GOTO9070
9130 REM
9140 ER=0:LA=LEN(A$):IFLA<3THEN9165
9145 PI=0:PF=0:FORIA=1TOLA:CH$=MID$(A$,IA,1):IFCH$="("THENPI=IA
9150 IFCH$=")"THEN9185
9155 NEXTIA:IFPI>0THENER=4:GOTO 8625
9160 FO$=A$:GOTO9065
9162 A$=FO$
9165 IFVAL(A$)=0THENA$="0":GOTO 8625
9175 IFLEFT$(A$,1)="+"THENA$=MID$(A$,2,80)
9180 GOTO 8625
9185 IFPI=0THENER=4:GOTO 8625
9190 PF=IA:FO$=MID$(A$,PI+1,PF-PI-1):EG=1:GOTO9065
9192 EG=0:IFER<>0THEN 8625
9195 IFPI>3THENGOSUB9220
9200 IFPI<=1THEN9210
9205 IFA<0ANDMID$(A$,PI-1,1)="-"THENPI=PI-1:FO$="+"+MID$(FO$,2,80)
9210 IFER<>0THEN 8625
9215 A$=MID$(A$,1,PI-1)+FO$+MID$(A$,PF+1,80):GOTO9140
9220 REM
9225 ER=0:CH$=MID$(A$,PI-1,1):IFCH$<"A"ORCH$>"Z"THENRETURN
```

```
9230 FU$=MID$(A$,PI-3,3):FORI=1TO6:IFFU$=FZ$(I)THEN9240
9235 NEXTI:ER=5:RETURN
9240 A1=VAL(FO$):ON I GOSUB 9245,9250,9255,9260,9265,9275:CA$=STR$(A1)
9241 IFA1>=0THENCA$="+"+MID$(CA$,2,80)
9242 EP=1:GOTO9280
9243 FO$=CA$:PI=PI-3:RETURN
9245 A1=ABS(A1):RETURN
9250 A1=INT(A1):RETURN
9255 A1=COS(A1):RETURN
9260 A1=SIN(A1):RETURN
9265 IFA1<0THENER=6:RETURN
9270 A1=SQR(A1):RETURN
9275 A1=TAN(A1):RETURN
9280 REM
9290 L=LEN(CA$):FORIC=1TOL:IFMID$(CA$,IC,1)="E"THEN9300
9295 NEXT:IFEP=1THENEP=0:GOTO9243
9297 GOTO9127
9300 N$=MID$(CA$,2,IC-2):L=LEN(N$)
9305 FORI=1TOL:IFMID$(N$,I,1)="."THENP=I:GOTO9315
9310 NEXT:P=L
9315 E=VAL(MID$(CA$,IC+1,3)):IFE<0THEN9355
9320 DD=L-P:IFDD=0THEN9345
9325 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80)
9330 IFDD<=ETHEN9345
9335 N$=MID$(N$,1,P+E-1)+"."+MID$(N$,P+E,80):DD=E
9345 N$=N$+MID$(ZE$,1,E-DD):GOTO9375
9355 IFP=LTHENP=P+2-L:GOTO9365
9360 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80):L=1
9365 N$=MID$(ZE$,1,-E-L+1)+N$
9370 N$=MID$(N$,1,P-1)+"."+MID$(N$,P,80)
9375 CA$=LEFT$(CA$,1)+N$:IFEP=1THENEP=0:GOTO9242
9377 GOTO9127
9900 REM *** ERRORES FICHEROS ***
9902 REM
9905 X=10:Y=24:GOSUB11000:PRINT"‾┌─┌─ EN UNIDAD DISCO";:GOTO 9910
9906 X=5:Y=22:GOSUB11000:PRINT"‾┌─┌─ , EL FICHERO DATOS YA SE"
9907 Y=Y+1:GOSUB11000:PRINT"HA CREADO EN ESTE DISCO":GOTO 9910
9908 X=4:Y=24:GOSUB11000:PRINT"‾┌─┌─ , EL FICHERO DATOS NO ESTA";:GOTO 9910
9910 X=12:Y=20:GOSUB11000:PRINT"PULSAR UNA TECLA "
9912 CLOSE1:CLOSE2:CLOSE15:REM CIERRE DE CANALES
9915 POKE198,0:WAIT198,1
9920 POKE198,0:RETURN
10000 REM ************ DATA ************
10001 DATA 145,17,29,157,140,133,136,13,25,134,135,0,0,0,0
10005 REM
10007 DATA"
10010 DATA"
10012 DATA"
10014 DATA"
10016 DATA"
10018 DATA"        '1'
10020 DATA"        '3'
10022 DATA"        '5'
10024 DATA"        '7'
10026 DATA"        '8'
10028 DATA"
10030 DATA"
10040 DATA"            DICIONES. ORUM, ","                ",5
10050 DATA"NTRODUCCION DATOS","RDENADO"
10060 DATA"USQUEDA CON VARIACION","ISCO DIRECTORIO","MPRESION"
10070 DATA"↑","/","*","-","+","ABS","INT","COS","SIN","SQR","TAN"
10200 REM ******************************
11000 REM
11001 REM **** POSICIONADO EN X,Y ****
11005 POKE 211,X
11010 POKE 214,Y
11015 SYS58732
11020 RETURN
12000 REM *** DIRECTORIO ***
12002 PRINTHO$;CHR$(142)
12005 X=1:Y=7:GOSUB11000:PRINT"INTRODUCIR EL DISCO Y PULSAR UNA TECLA"
12007 POKE198,0:WAIT198,1
12010 OPEN15,8,15,"IO":INPUT#15,AD,X$,CD,DD:IF AD<>0 THEN GOSUB20000:RETURN
12015 PRINTHO$:X=2:Y=3:GOSUB11000:PRINT" DIRECTORIO : ":PRINT
12020 CLOSE15:Y=4:OPEN 1,8,0,"$0"
12022 FORI=1TO4:GET#1,A$:NEXTI
12024 GET#1,A$:IFA$=""THENA$=CHR$(0)
12026 GET#1,B$:IFB$=""THENB$=CHR$(0)
12028 PRINTASC(A$)+256*ASC(B$);
12030 GET#1,A$
12032 IF A$=""THENPRINT:Y=Y+1:GOTO 12036
12034 PRINTA$;:GOTO 12030
12036 IF Y=20 THENGOSUB12100:Y=0:PRINT
12038 GET#1,A$:GET#1,B$:IF A$<>""ORB$<>""THEN 12024
12040 CLOSE1:DR=1
12100 PRINTTAB(6);"PULSAR UNA TECLA":POKE198,0:WAIT198,1
12110 IF DR<>0 THEN DR=0:PRINTCHR$(14);
12120 RETURN
13000 REM *** AYUDA ***
13005 CO=X:CV=Y
13010 X=1:Y=23
13015 GOSUB11000:PRINTH$(1);"      ";
13017 FOR HE=2 TO 12
13020 GETR$:IFR$<>CHR$(13)THEN13020
13022 GOSUB11000:PRINTH$(HE)
13030 NEXT HE
13035 GETR$:IFR$<>CHR$(13)THEN13035
13037 X=0:Y=23:GOSUB11000:PRINTSP$;
13040 X=2:Y=23:GOSUB11000:PRINT"EGISTRO N.";H;
13042 X=26:GOSUB11000:PRINT"";
13045 X=CO:Y=CV
13050 RETURN
14000 REM
14010 REM BUSQUEDA CON VARIACION
14030 PRINTHO$:X=1:Y=0:GOSUB11000:PRINTH1$
14040 X=9:Y=5:GOSUB11000:PRINTVO$(3)
14060 X=5:Y=12:GOSUB11000:PRINT""
14070 X=5:Y=6:GOSUB11000:PRINT"──────────────────────────"
14071 Y=14:GOSUB11000:PRINT"  0  = BUSCA EN EL CAMPO CLAVE "
14072 Y=16:GOSUB11000:PRINT"  -1  = VUELVE AL MENU "
14080 X=16:Y=20:GOSUB11000:INPUTA$:IFLEN(A$)>3THEN14030
14090 R=VAL(A$):IFR>VAL(DA$(0))THEN14030
14100 IFR<0THENRETURN
14110 IFR<>0THENW6=0:MIS=R:MAS=R:GOTO14250
14115 IFW6=2THEN14130
14120 PRINTHO$:X=2:Y=11:GOSUB11000:PRINT"PARA ESTE TIPO DE BUSQUEDA PRIMERO":Y=1
3:GOSUB11000
14122 PRINT"HAY QUE REALIZAR EL ORDENADO":FORI=1TO3000:NEXT:GOTO14000
14130 X=0:FORY=10TO21:GOSUB11000:PRINTLEFT$(SP$,40);:NEXT
14140 X=2:Y=11:GOSUB11000:PRINT""
14150 X=2:Y=15:GOSUB11000:PRINT": EL CAMPO CLAVE DEBE SER EL "
14152 X=8:Y=16:GOSUB11000:PRINT"MISMO QUE EL DEL ORDENADO."
14160 X=6:Y=20:GOSUB11000:INPUTRA$
14162 IFLEN(RA$)>LC(K4)THEN14130
14165 PRINTHO$:X=8:Y=6:GOSUB11000:PRINT"ESPERAR, BUSQUEDA DE  :":X=(40-LEN(RA$))
/2
14166 Y=8:GOSUB11000:PRINTRA$
14170 IFTP(K4)=3THENRA$=RA$+MID$(SP$,1,LC(K4)-LEN(RA$)):GOTO14190
14180 RA$=MID$(SP$,1,LC(K4)-LEN(RA$))+RA$
14190 GOSUB14500:REM BUSQUEDA BINARIA
14195 FORI=1TO1000:NEXT
14200 X=1:Y=6:GOSUB11000:PRINTLEFT$(SP$,38)
14210 IFTR=0THENX=10:Y=13:GOSUB11000:PRINT""
14211 IFTR=0THENFORI=1TO1000:NEXT:GOTO14000
14220 A=MAS-MIS+1:X=13:Y=13:GOSUB11000:PRINT"":X=11:Y=19:GOSUB11000
14230 IFA=1THENPRINT"(UN SOLO REGISTRO)":GOTO14240
14235 PRINT"(";A;" REGISTROS )"
14240 FORI=1TO2000:NEXT
14250 REM PRESENTA REGISTRO
14260 FORJK=MISTOMAS:KJ=IS(JK)
14265 IFR<>0THENKJ=JK
14270 A$=MID$(DA$(KJ),2,LEN(DA$(KJ))-1)
14280 C1=1:FORI=1TOM3:IFTP(I)=0THEN14300
14290 B$(I)=MID$(A$,C1,LC(I)):C1=C1+LC(I):HTP(I)=TP(I):TP(I)=4
14300 NEXTI
14310 H=JK:IFR=0THENH=KJ
14320 GOSUB2000:X=2:Y=23:GOSUB11000:PRINT"EGISTRO N.";H
14325 X=26:GOSUB11000:PRINT"";
14330 FORI=1TOM3:TP(I)=HTP(I):NEXT:IFR=0THENTP(K4)=4
14340 GOSUB3000:TP(K4)=HTP(K4)
14350 IF FS=6THEN14270
14352 IFFS=7THENRETURN
14360 DA$(KJ)="*"
14370 FORI=1TOM3:IFTP(I)=0THEN14390
14380 DA$(KJ)=DA$(KJ)+B$(I)
14390 NEXTI
14420 NEXTJK:X=0:Y=23:GOSUB11000:PRINTSP$;
14421 X=9:Y=23:GOSUB11000:PRINT"OTRA VARIACION? (S/N)";
```

```
14430 GETA$:IFA$="S"THEN14000
14440 IFA$<>"N"THEN14430
14450 GOSUB5000:RETURN
14460 REM
14500 REM BUSQUEDA BINARIA
14515 P1=P8:P2=P9
14520 RI=0:RF=VAL(DA$(0)):RM=INT(RF/2)
14530 RE=IS(RM):RB$=MID$(DA$(RE),P1,P2)
14540 IFRA$=RB$THEN14640
14550 IFRA$>RB$THEN14600
14560 RF=RM:RM=INT((RF-RI)/2)
14570 IFRM<1THENRM=1
14580 RM=RM+RI:IFRM>=RFTHENTR=0:RETURN
14590 GOTO 14530
14600 RI=RM:RM=INT((RF-RI)/2)
14610 IFRM<1THENRM=1
14620 RM=RM+RI:IFRM>RFTHENTR=0:RETURN
14630 GOTO14530
14640 SG=-1:RF=RM:RI=RM:MIS=RM:MAS=RM:FMI=0
14650 RM=RM+SG:RE=IS(RM):RB$=MID$(DA$(RE),P1,P2)
14660 IFRA$=RB$THEN14690
14670 IFFMI=1THENTR=1:RETURN
14680 FMI=1:SG=1:RM=RF:GOTO14650
14690 IFFMI=1THENMAS=MAS+1:GOTO14700
14695 MIS=MIS-1
14700 GOTO14650
16000 REM
16001 REM          IMPRESION
16002 REM
16020 IF W6=2 THEN OPEN4,4:GOTO16045
16035 PRINTHO$:X=1:Y=9:GOSUB11000:PRINT"ANTES DE LA IMPRESION HAY QUE HACER"
16037 Y=Y+2:GOSUB11000:PRINT"EL ORDENADO. OPCION NO REALIZABLE"
16040 FOR I=1 TO 5000:NEXTI:RETURN
16045 GOSUB600:CMD4:GOSUB17000:CT=4
16050 FORJ=1TOM3:Z8(J)=0:NEXT J
16055 FORI=1TOMR
16060 IF IS(I)=0 THEN 16200
16065 IFCT>40THENFORKK=1TO65-CT:PRINT:NEXTKK:CT=4:GOSUB17000
16070 A$=MID$(DA$(IS(I)),2,200)
16075 C1=1:FORJ=1TOM3:IFTP(J)=0THEN16100
16080 B$(J)=MID$(A$,C1,LC(J)):C1=C1+LC(J)
16100 NEXTJ
16105 IFFI=2THENPRINT"────────────":PRINT"REGISTRO N.";IS(I):PRINT"──────────
──"
16110 A$="|":FORJ=1TOM3:IFNS(J)=0THEN16150
16120 IFFT(J)=1THENZ8(J)=Z8(J)+VAL(B$(NS(J)))
16130 IF FI=1THENA$=A$+B$(NS(J))+"|":GOTO16150
16140 PRINTB$(NS(J)):CT=CT+1
16150 NEXTJ
16160 IFFI=1THENPRINTA$:CT=CT+1:GOTO16170
16165 PRINTS1$:CT=CT+4
16170 NEXTI
16200 REM TOTALES
16210 PRINTTAB(0);S1$
16220 IFFI=2THEN16300
16230 A=1:FORI=1TOM3:IFTP(I)=0THEN16260
16240 A=A+LC(NS(I))+1:IFFT(I)=0THEN16260
16245 AA=A-1-LEN(STR$(Z8(I))):IFAA<0THENAA=0
16250 PRINTTAB(AA);Z8(I);
16260 NEXT I:PRINT
16280 GOTO 16355
16300 PRINTTAB(4);"| ┌ | ♠ L ‾ ● :":PRINTTAB(4)"────────────"
16310 FORI=1TOM3:IFTP(I)=0THEN16340
16320 IFFT(I)=0THEN PRINT:GOTO16340
16330 PRINTTAB(4);"CAMPO :";I;"=";Z8(I)
16340 NEXT I
16355 FORKK=1TO64-CT:PRINT:NEXTKK:PRINT#4:CLOSE4:RETURN
17000 REM
17005 PRINTCHR$(17)
17010 S1$="":FORIK=1TO80:S1$=S1$+"-":NEXTIK
17020 PRINTS1$:PRINT
17025 PRINTTAB(XP);II$
17030 PRINTTAB(0);S1$
17050 RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - COMMODORE 64

Definiciones

D$(∗)	= descripciones
TP(∗)	= tipos de los campos
LC(∗)	= longitud campo de introducción
ND(∗)	= número de decimales
XD(∗) **YD(∗)**	= coordenadas en el vídeo de las descripciones
XC(∗) **YC(∗)**	= coordenadas de los campos
FO$(∗)	= matriz de las fórmulas de cálculo
FT$(∗) **NS$(∗)**	= parámetros de impresión
L1$,H1$ **L2$**	= cabecera
A	= código de la tecla pulsada
XP	= columna principio impresión
II$	= cabecera de impresión

Main

B$(∗)	= matriz de los buffers (1 por campo de introducción)
LC(∗)	= longitud del campo de introducción
ND(∗)	= número de decimales en el campo de introducción
CX(∗) **CY(∗)**	= coordenadas en el vídeo del campo de introducción
DX(∗) **DY(∗)**	= coordenadas en el vídeo de las descripciones
TF(∗)	= matriz numérica que contiene los códigos ASCII de las teclas de comando
D$(∗)	= descripciones en las máscaras vídeo
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de introducción alfanuméricos
NS$	= carácter "#"; identifica los campos de introducción numéricos
NUC	= número de campos de introducción en la máscara vídeo activada
MR	= número máximo de registros
HT(∗) **HL(∗)**	= matriz auxiliar para máscara vídeo ordenado
HN(∗) **IS(∗)**	= indicadores de los registros después del ordenado
NS(∗) **FT(∗)**	= parámetros de impresión
Z8(∗)	= totales en fase de impresión
F$(∗)	= matriz de las fórmulas de cálculo
DA$(∗)	= matriz de los registros
S9(∗, ∗)	= índice para ordenado
KJ	= índice del bucle de introducción
MX	= último registro introducido
K4	= número del campo clave del ordenado
ERR	= indicador de error
OP$	= operadores reconocidos (↑,/,∗,–,+)
FZ$	= funciones reconocidas (ABS,INT,COS,SIN,SQR,TAN)

La lógica del programa

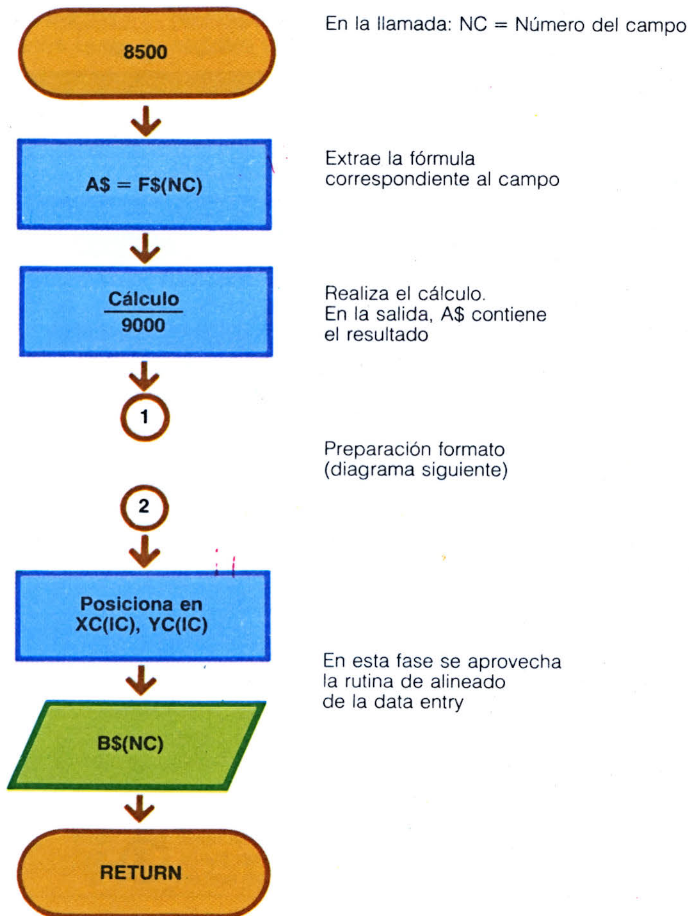
En el programa se ha previsto la posibilidad de definir un campo como resultado de procesos, lo que implica una ulterior modificación de la data entry. En la versión anterior, al encontrar un campo definido de cálculo se activa una subrutina con la que se indica expresamente el proceso a realizar. Esta estructura no puede utilizarse en la Base de Datos porque no permite introducir fórmulas «del exterior».

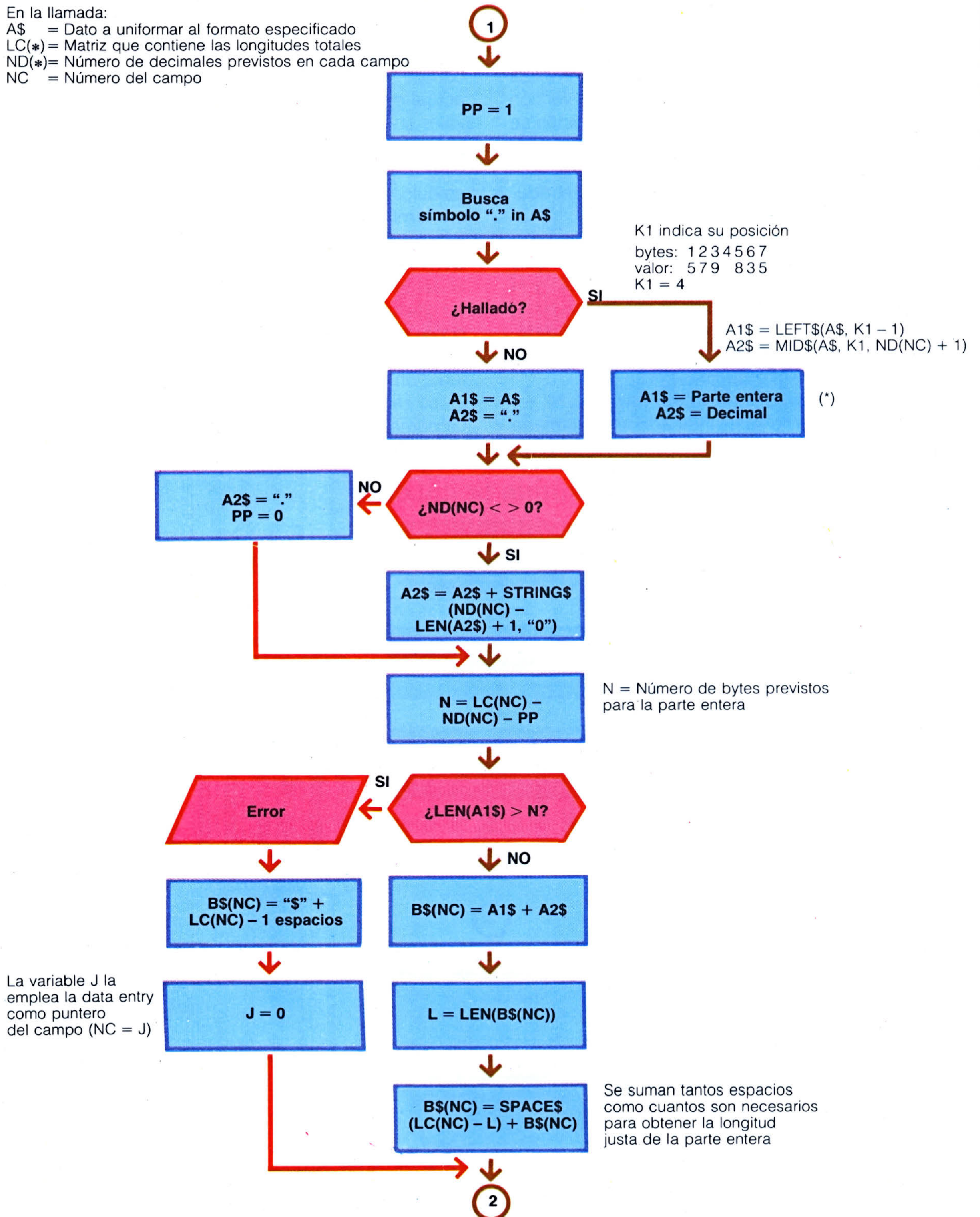
En consecuencia, se ha incluido una gestión análoga a la utilizada en el Tablero electrónico: las fórmulas a aplicar están memorizadas en la matriz F$(20) y la subrutina de cálculo debe extraer la correspondiente al campo actual para «pasarla» a la subrutina de desarrollo. En la salida, el valor resultante se transfiere al campo, o sea al elemento de B$(*).

El diagrama de principio de esta subrutina se ha representado abajo. Como se ha indicado, en la llamada hay que implantar en la variable NC el número del campo actual, o sea el correspondiente a la posición actual del cursor en la máscara vídeo. Utilizando este valor como puntero se extrae la fórmula (de la matriz F$(*), construida en la fase de generación) y se transfiere a la variable A$, como lo indica la subrutina de cálculo. En la salida, esta última restituye el resultado en la misma variable A$, que sólo debe adaptarse al formato implantado en la generación y transferirse a B$(NC).

En la fase de generación de la máscara, cada campo de cálculo se indica escribiendo tantos símbolos "C" como cuantas cifras previstas hay; por tanto, es necesario uniformizar el resultado del cálculo, que podría tener un formato diferente. Por ello, después del cálculo, se llama la subrutina de la página siguiente.

La primera función que realiza es la búsqueda del sím-




PREPARACION DEL FORMATO
En la llamada:
A$ = Dato a uniformar al formato especificado
LC(*) = Matriz que contiene las longitudes totales
ND(*) = Número de decimales previstos en cada campo
NC = Número del campo
1
PP = 1
Busca símbolo "." in A$
K1 indica su posición
bytes: 1 2 3 4 5 6 7
valor: 5 7 9 8 3 5
K1 = 4
¿Hallado?
SI
A1$ = LEFT$(A$, K1 – 1)
A2$ = MID$(A$, K1, ND(NC) + 1)
NO
A1$ = A$
A2$ = "."
A1$ = Parte entera
A2$ = Decimal
(*)
A2$ = "."
PP = 0
NO
¿ND(NC) < > 0?
SI
A2$ = A2$ + STRING$ (ND(NC) – LEN(A2$) + 1, "0")
N = LC(NC) – ND(NC) – PP
N = Número de bytes previstos para la parte entera
Error
SI
¿LEN(A1$) > N?
NO
B$(NC) = "$" + LC(NC) – 1 espacios
B$(NC) = A1$ + A2$
La variable J la emplea la data entry como puntero del campo (NC = J)
J = 0
L = LEN(B$(NC))
B$(NC) = SPACE$ (LC(NC) – L) + B$(NC)
Se suman tantos espacios como cuantos son necesarios para obtener la longitud justa de la parte entera
2
(*) La segunda transferencia sólo se hace si ND(NC) < > 0

bolo "." para poder determinar cuál es la parte entera y cuál la decimal.
Al final de la búsqueda pueden presentarse dos casos:

1 / El símbolo "." no se ha hallado
2 / El símbolo "." se ha hallado

En el primer caso, A$ sólo contiene un valor entero que se transfiere a la variable A1$, mientras que, en el segundo, la parte entera de A$ está separada de la decimal, originando las dos variables A1$ y A2$.
Las longitudes de las dos variables (teniendo en cuenta que en el primer caso A2$ es nula) se comparan después con las longitudes asignadas al campo (LN(*) como valor total y ND(*) como parte decimal). En base al resultado de esta comparación, las dos longitudes se adaptan a lo especificado en el formato, siguiendo las reglas:

- La parte entera no puede tener una longitud mayor de la prevista, por lo que si se produce esta eventualidad, el contenido del campo es sustituido por el símbolo $, que indica error
- En el caso contrario, parte entera de longitud menor a la prevista, el campo se completa con espacios en blanco a la izquierda del valor (recuérdese que un número debe estar en el lado derecho).

En cambio, para la parte decimal, la lógica es la siguiente:

- En el caso de longitud superior a la prevista se truncan las cifras de la derecha (las menos significativas)
- En cambio, si la longitud es inferior al campo, se completa colocando ceros a la derecha

Al final de estos procesos, el primero sobre la cadena A1$ y el segundo sobre la A2$, se reconstruye el campo sumándolas. En esta versión no se realiza ningún redondeo, por lo que la parte decimal simplemente se trunca y la parte entera indica exactamente el resultado del cálculo. En cambio, en algunas aplicaciones es útil disponer de valores redondeados. En estos casos, hay que modificar el programa incluyendo los redondeos antes de la reconstrucción de la cadena A$.

Estructura de los ficheros. La modularidad y la flexibilidad del programa implican una estructura de ficheros relativamente compleja. Efectivamente, además de los datos propiamente dichos, que constituyen el archivo, hay que memorizar una serie de informaciones accesorias. En particular, los valores a memorizar son:

- Fórmulas
- Descripciones y parámetros de la máscara
- Cabeceras de los tabulados
- Formatos de impresión
- Índice para la extracción de los datos ordenados (Sort)

Las principales características de cada fichero se han representado en la tabla que puede verse abajo.
En las páginas 2472 y 2473 se han indicado los formatos de fichero utilizados. Estos ficheros, completamente transparentes para el usuario, son generados por el sistema durante la definición del archivo y tienen nombres, longitudes y formatos predefinidos y no modificables desde el exterior, o sea con ficheros de «servicio» utilizados exclusivamente por el sistema.
Al contrario del fichero datos, el que deberá contener las informaciones que el usuario desea procesar tiene un nombre que el usuario debe definir y un formato de registro variable en función de la máscara vídeo.

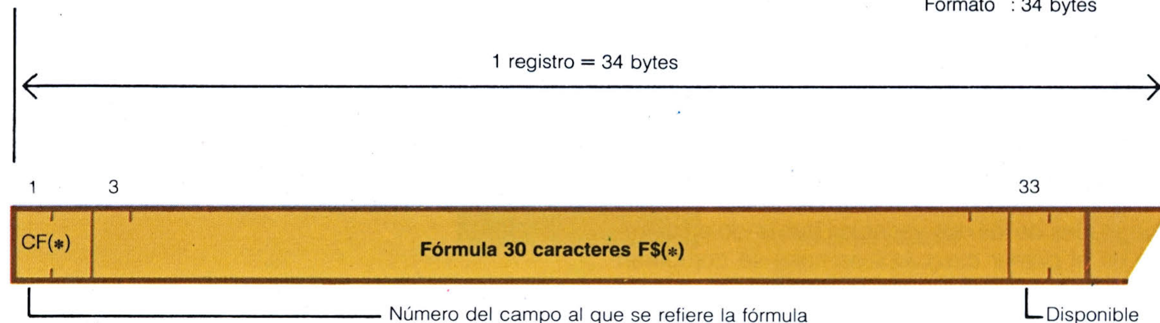
Fichero de las fórmulas. La estructura de este fichero es una de las más sencillas. Cada registro contiene una de las 20 posibles fórmulas, de 30 bytes de longitud máxima.
En el formato del registro se han incluido además dos bytes iniciales en los que se memorizan el número del campo a que se refiere la fórmula y dos bytes finales disponibles para eventuales implantaciones. Por tanto, resulta una longitud de registro de 34 bytes y un total de 20 registros, uno para cada campo. El fichero tiene formato fijo, por lo que los eventuales registros no utilizados deben escribirse en blanco. Además, la transferencia al disco es de acceso directo en función del número del campo. Por tanto, la fórmula correspondiente al primer campo ocupará el registro 1, la del segundo campo el registro 2, etc.
Esta lógica hace inútil la memorización del número del campo a que se refiere la fórmula; de hecho, este valor lo indica el mismo número de registro. No obstante, los

FICHEROS DE SERVICIO UTILIZADOS

Nombre	Registros	Byte	Ocupación	Contenido
FOR @	20	34	680	Fórmulas
DEF @	20	26	520	Descripciones de la máscara
PIN @	1	82	82	Línea de cabecera de impresión
PDF @	1	82	82	Indice de los campos a imprimir

FORMATO DE REGISTRO DEL FICHERO DE FORMULAS

Nombre : FOR @
Registros : 20
Formato : 34 bytes



Instrucciones de extracción

CF(*) = LEFT$(A$,2)
F$(*) = MID$(A$, 3,30)
Disponible = RIGHT$(A$,2)

dos primeros bytes están reservados para este objeto, incluso si su contenido es igual al número del registro; de esta manera es posible disponer de un puntero útil para eventuales implantaciones.

Fichero de las definiciones. También este fichero contiene 20 registros, uno por cada uno de los campos previstos que deben estar presentes, incluso si se dejan en blanco.
El formato de cada registro (26 bytes) prevé una estructura en la que a cada descripción corresponde un campo de datos y, viceversa, para cada campo de datos se proporciona la descripción. Esta lógica no excluye la posibilidad de utilizar un número de descripciones distinto del de los campos: efectivamente, en este caso, la parte del registro correspondiente al parámetro no utilizado deberá dejarse en blanco. Como alternativa a esta estructura, el fichero podría haberse dividido en dos partes, cada una con 20 registros: la primera correspondiente sólo a las descripciones y la segunda para las definiciones de las áreas de datos. Sin embargo, de esta manera se tendría un desaprovechamiento de espacio, puesto que el registro de las descripciones debería ser más largo que el de parámetros, obligando a completar este último con un cierto número de espacios en blanco.
Una segunda alternativa podría permitir el empleo de un fichero secuencial y, por tanto, con registros de longitud variable: sin embargo, de esta manera se tendría una estructura menos elástica y no homogénea.

Fichero de la cabecera. En la gestión de los archivos, una de las funciones principales es la impresión de su contenido y, en esta fase, es particularmente útil disponer de una cabecera para indicar el principio de cada página.
Para ello se ha previsto un fichero, de un solo registro, cuyo contenido se introduce en la fase de definición de los formatos de impresión y que se representará automáticamente al principio de cada página.
Además, en el registro se ha previsto un campo de 2 bytes que especifica la columna de principio de impresión de la cabecera.

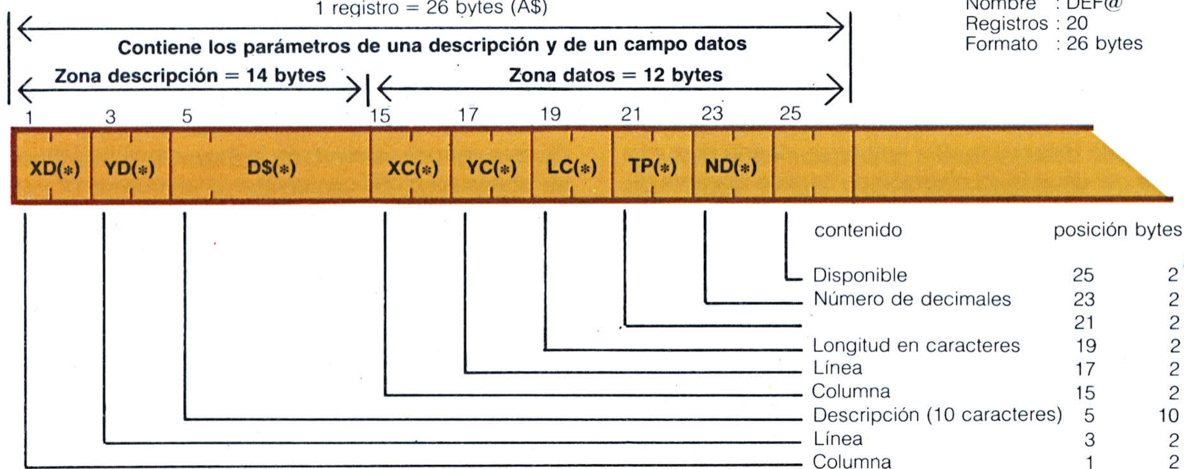
Fichero de los formatos de impresión. En la fase de impresión no siempre es necesario representar todos los campos previstos en el fichero datos, e incluso un tabulado demasiado complejo y que contenga informaciones no necesarias, puede ser causa de confusión o de interpretación errónea de los datos. Por este motivo, la subrutina de impresión es parametrizada, en el sentido de que es posible definir cuáles deben ser los datos a presentar y en qué posición. Además se ha previsto una función de totalización de los valores numéricos contenidos en los campos impresos. Por tanto, los parámetros asociados a la impresión son:

- Tipo de impresión
- Número del campo
- Flag que indica si el contenido debe ser totalizado o no

Estos valores pueden memorizarse en un fichero para un uso posterior, formado por un solo registro que contiene los datos anteriores, más un byte disponible, para cada uno de los 20 campos previstos.

FORMATO DE REGISTRO DEL FICHERO DEFINICIONES

Nombre : DEF@
Registros : 20
Formato : 26 bytes



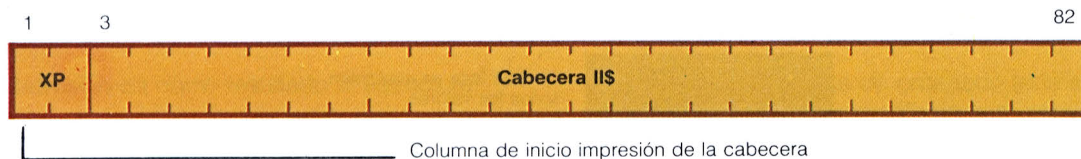
Instrucciones de extracción

XD(*) = LEFT$(A$,2)
YD(*) = MID$(A$,3,2)
D$(*) = MID$(A$,5,10)
XC(*) = MID$(A$,15,2)
YC(*) = MID$(A$,17,2)
LC(*) = MID$(A$,19,2)
TP(*) = MID$(A$,21,2)
ND(*) = MID$(A$,23,2)
Disponible = RIGHT$(A$,2)

N.B.: Las instrucciones son indicativas. Se ha omitido VAL..., que permite la transferencia de los datos a una variable numérica

CABECERAS DE IMPRESION

Nombre : PIN@
Registro : 1
Formato : 82 bytes



Instrucciones de extracción

XP = VAL(LEFT$(A$,2))
II$ = RIGHT$(A$,80)

FORMATO DE IMPRESION

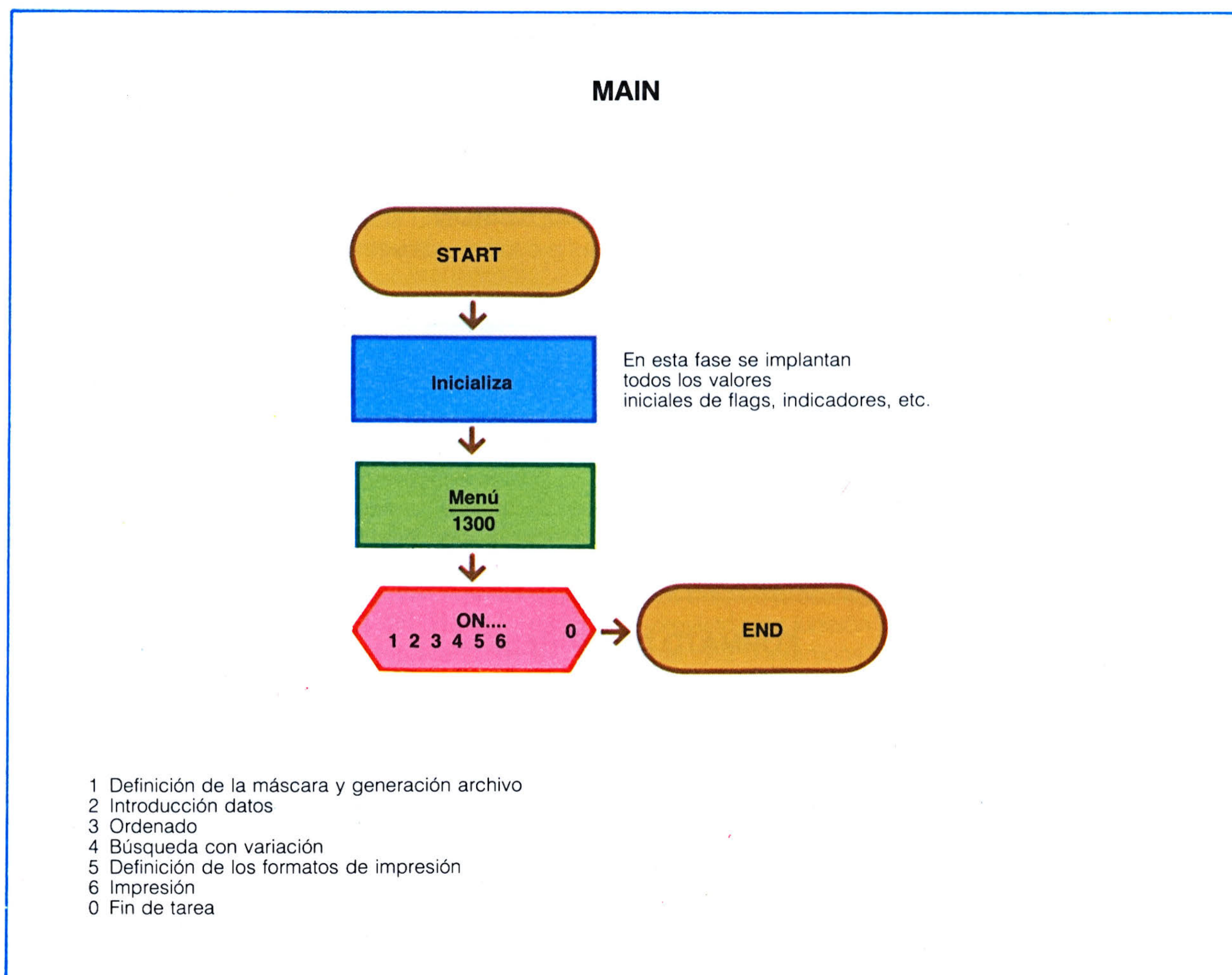
Nombre : PDF@
Registro : 1
Formato : 82 bytes



Menú principal. Al activar el programa se presenta el menú principal para la selección deseada. Abajo se ha representado el diagrama de principio.
Las funciones previstas (voces de 1 a 6) pueden realizarse con otras tantas subrutinas o con programas separados; la elección de uno u otro método depende del tipo de máquina, puesto que está vinculada a la velocidad de carga de los programas y a la cantidad de memoria disponible. Sin embargo, desde el punto de vista del usuario, la elección no implica ninguna diferencia y, en lo que respecta a la escritura del programa, se reduce a sustituir las instrucciones GOSUB... por RUN (o equivalente).
Sin embargo, hay que tener en cuenta que la elección, si bien es muy sencilla en términos de instrucciones a sustituir, en realidad implica importantes modificaciones de la lógica; efectivamente, en el caso de que se estructuren subrutinas, todos los parámetros o las variables pueden ser utilizadas por cualquier voz del menú, mientras que si se convierten algunas en forma de programa es necesaria una nueva inicialización. Esto no obstante, la solución de segmentación en varios programas es preferible, en particular para las funciones de Ordenado. De hecho, para realizar más rápidamente un Sort (ordenado), es necesario disponer de amplias áreas de memoria a la que transferir los datos.
Para las funciones de impresión también pueden adoptarse soluciones de un programa en sí mismo, pero por motivos diferentes a los anteriores: la impresión no requiere amplias zonas de memoria y constituye una función bien separable del resto. Las ventajas de esta estructura están en la mayor facilidad de aportar al mismo modificaciones e implantaciones y en la posibilidad de utilizar las funciones así separadas como programas de utilidad general para otras aplicaciones. La estructura segmentada puede adoptarse tanto para la versión de disco como para la de cinta, aunque en este último caso, el usuario debe operar en la unidad de cinta para posicionarla cada vez que el programa lo pide y, por tanto, no es funcional.
Como ya se ha indicado, las dos funciones que pueden separarse del resto del programa son:

1 / Ordenado
2 / Impresión

Ambas necesitan algunos parámetros y valores implantados en el main, por lo que deben contener una



copia de las subrutinas que atribuyen estos valores. En particular, la más importante es la adquisición del formato del fichero datos.
En la generación, después de haber definido la máscara vídeo y de haber leído sus características, el sistema transfiere los valores a los distintos ficheros previstos en el procedimiento y si el programa no se interrumpe, no es necesario nada más para la activación de las distintas funciones. Segmentando algunas voces en el momento de su carga, los parámetros se ponen a cero, por lo que es necesario recargarlos en la memoria leyéndolos en los ficheros anteriores.
En realidad, no todos los segmentos deben utilizar los mismos ficheros, por lo que pueden preverse instrucciones de carga distintas en función de la actividad que debe realizar el módulo. De hecho, se tiene:

Fichero	**Función que lo utiliza**
FOR @	Data entry
DEF @	I/O disco, Data entry, Sort, Búsqueda, Impresión
PIN @	Impresión
PDF @	Impresión

Por tanto, las lecturas de los ficheros pueden agruparse como sigue:

- Todos en la fase de generación
- FOR, DEF para la voz 2 (introducción)
- FOR, DEF para Ordenado y Búsqueda
- FOR, PIN, PDF para Impresión

Esta subdivisión da como resultado la estructura:

- Un primer programa para la generación (voz 1)
- El segundo para la voz 2 (Introducción)
- El tercero para las voces 3 y 4
- El cuarto para las voces 5 y 6

Naturalmente, esta estructura puede activarse fácilmente en la versión disco, mientras que para la de cinta, las dificultades de carga de los distintos programas la hacen poco práctica. Además, tal y como se ha indicado anteriormente, la solución elegida depende del tipo de máquina utilizada, por lo que puede variar según la versión.

Definición de las máscaras. En la página siguiente se ha representado el diagrama del módulo predispuesto para la definición de la estructura de las máscaras y del fichero asociado. La lógica seguida puede aplicarse tanto al caso de estructura por subrutinas como al caso de programa en sí mismo; en particular, para esta función es más adecuada la segunda solución, ya que el módulo puede estar completamente separado del resto porque produce una salida (parámetros de la máscara) en el fichero y, por tanto, es accesible a otros programas.
El diagrama es en principio válido para todas las versiones, mientras que el detalle de su desarrollo está estrechamente ligado al tipo de máquina: en particular las dos subrutinas de gestión del vídeo (preparación y lectura) tienen estructuras e instrucciones no generalizables, por lo que deben consultarse los correspondientes listados.
Finalmente, hay que tener en cuenta que en el diagrama no se ha representado la función de preparación de los ficheros, que consiste en pedir su nombre, su longitud (estimada) en número de registros y en escribir en el disco de datos estos registros en blanco.

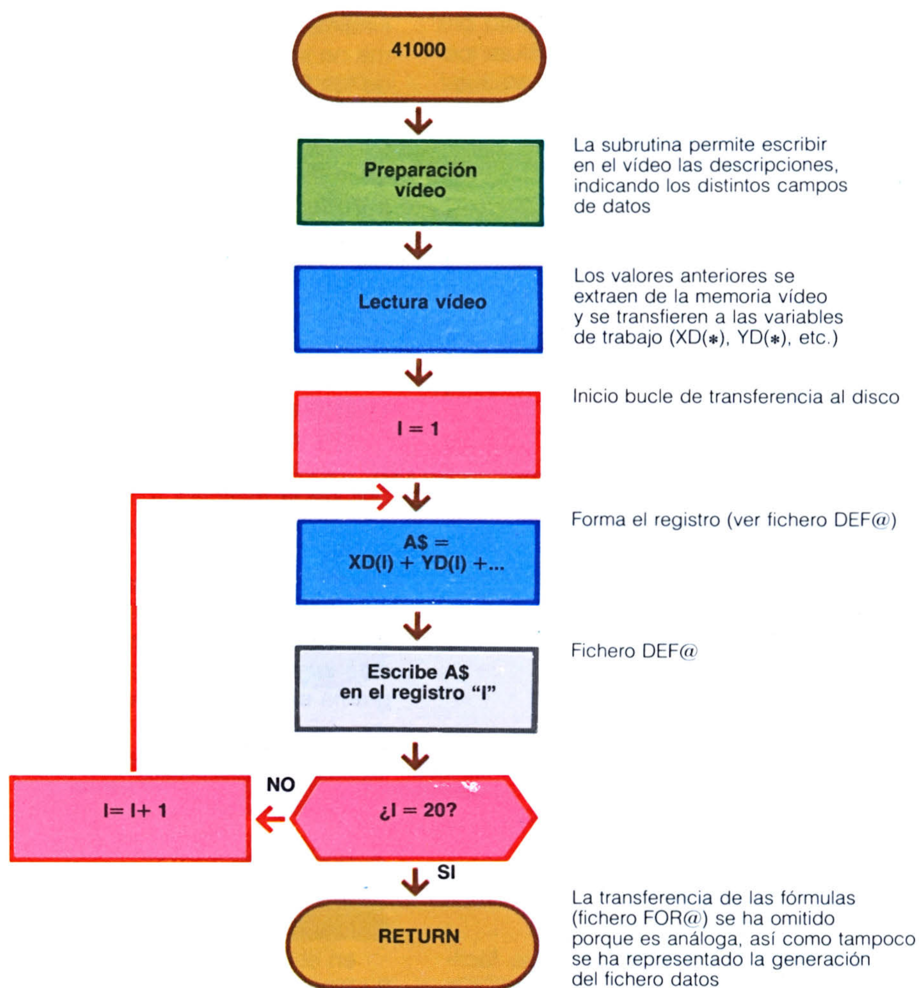
Introducción de datos. El desarrollo de esta función es del todo idéntico al desarrollo para otras aplicaciones, por lo que no necesita comentario, salvo llamar la atención sobre la presencia del directorio en el fichero datos: por tanto, como de costumbre, el dato número 1 se memoriza en el registro 2 y así sucesivamente.
Sin embargo, hay que hacer una observación. En las funciones previstas por el menú hay incluidos el ordenado y la búsqueda, basada en un fichero ordenado. En particular, el Sort, desarrollado en uno de los campos de la máscara (campo clave) pierde su validez con cada introducción; por tanto, el usuario debe reactivarlo.
Como alternativa puede utilizarse una estructura que prevea automáticamente el ordenado en cada nueva introducción (o actualización del campo clave), pero esta implantación haría mucho más complejo el programa sin ofrecer una ventaja determinante.

Ordenado. El programa de ordenado (o la subrutina, según la versión) se desarrolla en tres etapas:

1 / Elección, por parte del usuario, del campo a utilizar como clave, o sea del campo en base al cual se realizará el ordenado
2 / Determinación de los correspondientes punteros en el registro datos
3 / Sort, o sea preparación de la matriz IS(*), del índice del ordenado y su memorización

Terminada esta función, el fichero IND @ contiene los punteros a los registros en el orden que se deriva del campo clave elegido y puede emplearse como índice con el que obtener la impresión con los datos ordenados o para realizar una búsqueda rápida.
Como se ha indicado, la primera función a realizar es la elección, por parte del usuario, del campo clave en base al cual realizar el ordenado. El método indicado en el diagrama consiste en presentar una data entry con las descripciones de cada campo; el usuario debe indicar cuál es la clave elegida introduciendo en

DEFINICION MASCARAS

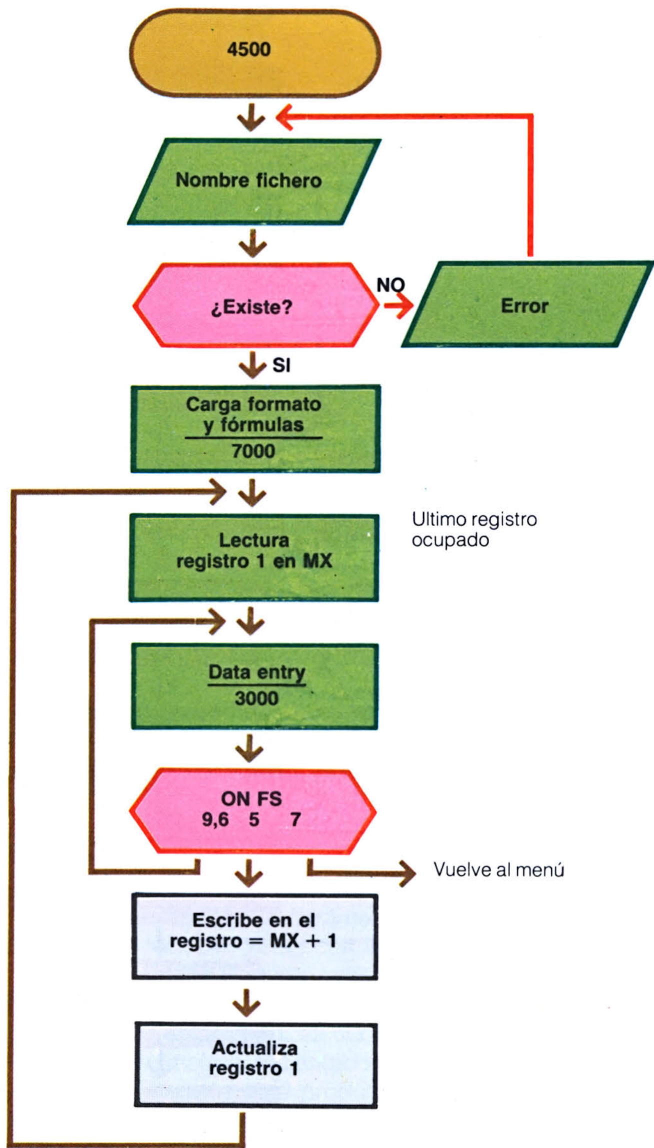


La subrutina de llamada necesita los valores XD(*), YF(*), etc. de acuerdo con el formato del fichero

(*) El bloque es indicativo, porque no se han representado las intrucciones de conversión de los números en cadenas ni el correspondiente control sobre las longitudes (ver tabla del fichero DEF@). Además, el programa permite gestionar un máximo de 20 descripciones y 20 campos de datos. En el caso de que el número utilizado fuese inferior, los registros no utilizados, siempre de 26 bytes de longitud, se dejan en blanco

correspondencia el símbolo "*". Para cada ordenado se extrae, como clave, sólo el primer campo que contiene "*", por lo que si la máscara también acepta más de una introducción, el elemento utilizado es sólo el primero.
Para obtener esta funcionalidad son necesarias algunas modificaciones, antes y después de la selección. Antes de utilizar esta máscara en particular, en la que todos los campos tienen longitud 1, hay que salvar los datos reales que describen el registro. Al final de la elección, los valores reales deben recuperarse y transferirse nuevamente a las respectivas variables (la recuperación no se ha indicado en el diagrama y se reduce a un bucle antes de la instrucción K4 = I).
Después de haber recuperado los valores precisos en las variables dimensionadas TP(*), LC(*) y ND(*), el programa calcula los bytes de inicio y fin del campo clave para extraer el correspondiente contenido durante el Sort. Las variables P1 y P2 del diagrama pueden tener nombres diferentes en las distintas versiones.
Al final del Sort, la correspondiente subrutina restituye la matriz IS(*) que contiene los punteros a los registros en el orden que se deriva del campo clave elegido, por lo que se debe memorizar dicho índice para utilizarlo después de las impresiones y en todas las funciones que requieren el fichero datos ordenado.
El uso de un fichero índice necesita algunas modificaciones de las funciones, porque cada dato se toma a través de un direccionamiento indirecto. Por ejemplo, si se quieren leer los datos ordenados, es necesario un bucle desde el valor 1 (primer dato) a MX (número de datos presentes) que extraiga el valor de SI(*) y lo utilice como número de registro.

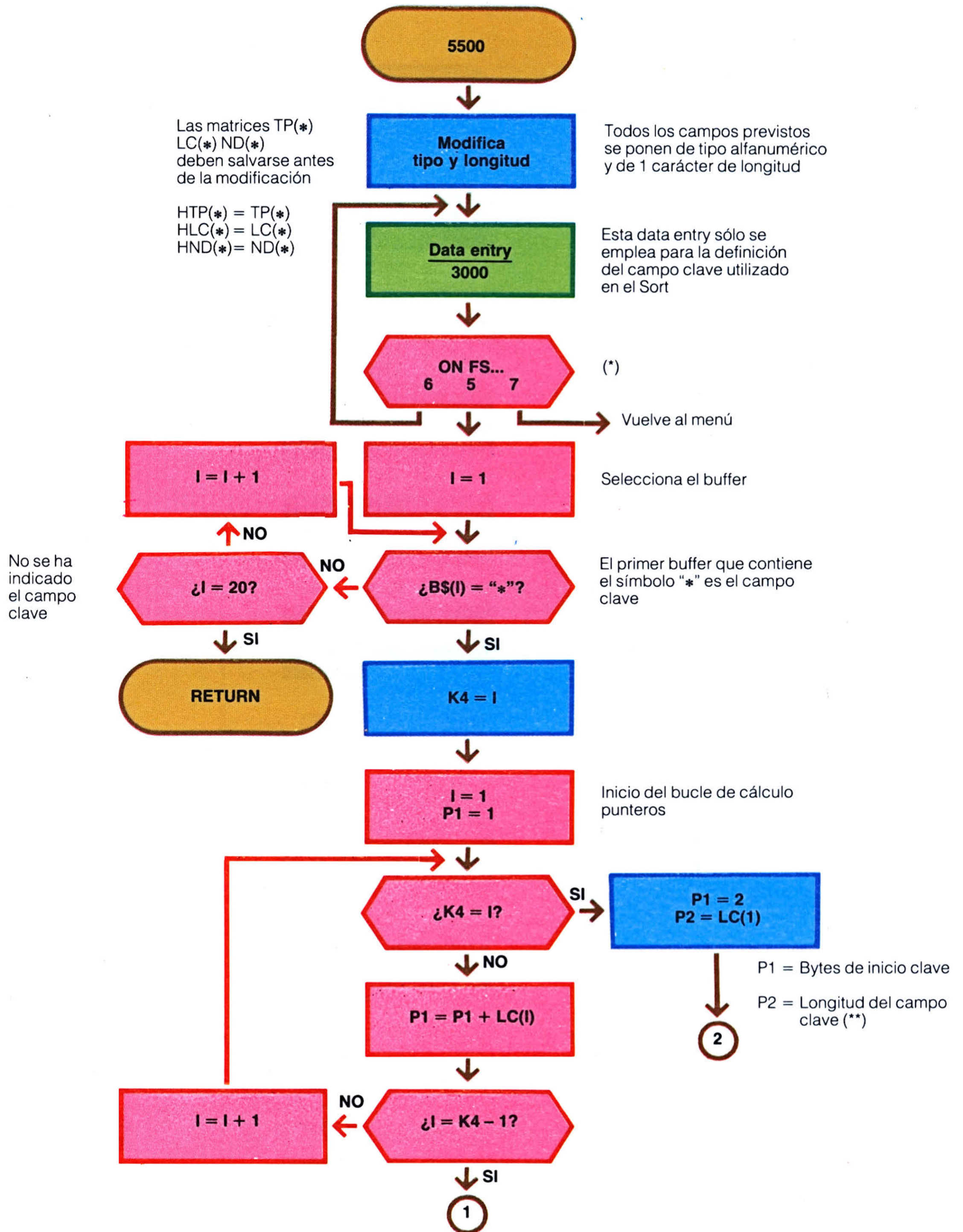
INTRODUCCION DE DATOS

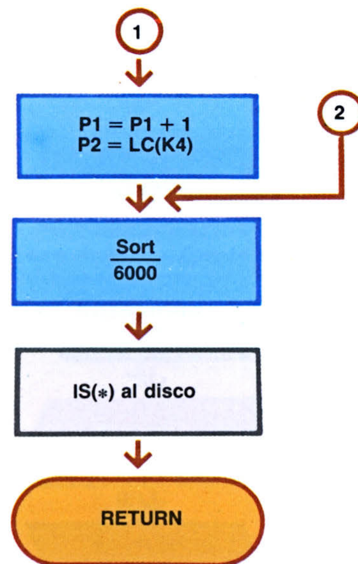


En la versión cinta (MSX), esta rutina se realiza de manera diferente, porque los datos se transfieren a la memoria en la matriz DA$(*) y sólo después de la última introducción se graban en la cinta (ver el listado)

ORDENADO

En la entrada deben proporcionarse todos los parámetros que definen la máscara vídeo





(*) En el caso de que FS = 5 y FS = 7, antes de proseguir hay que volver a los valores originales las matrices TP(*), LC(*), ND(*): TP(*) = HTP(*)

(**) El primer byte del registro es un asterisco ("*")

Esta lógica se muestra con mayor detalle en la subrutina de impresión.

Búsqueda con variación. La búsqueda y la eventual actualización de un registro pueden realizarse de dos maneras:

- Especificando su número, que corresponde al número progresivo de introducción
- Especificando el valor del campo clave según el cual debe ordenarse el fichero

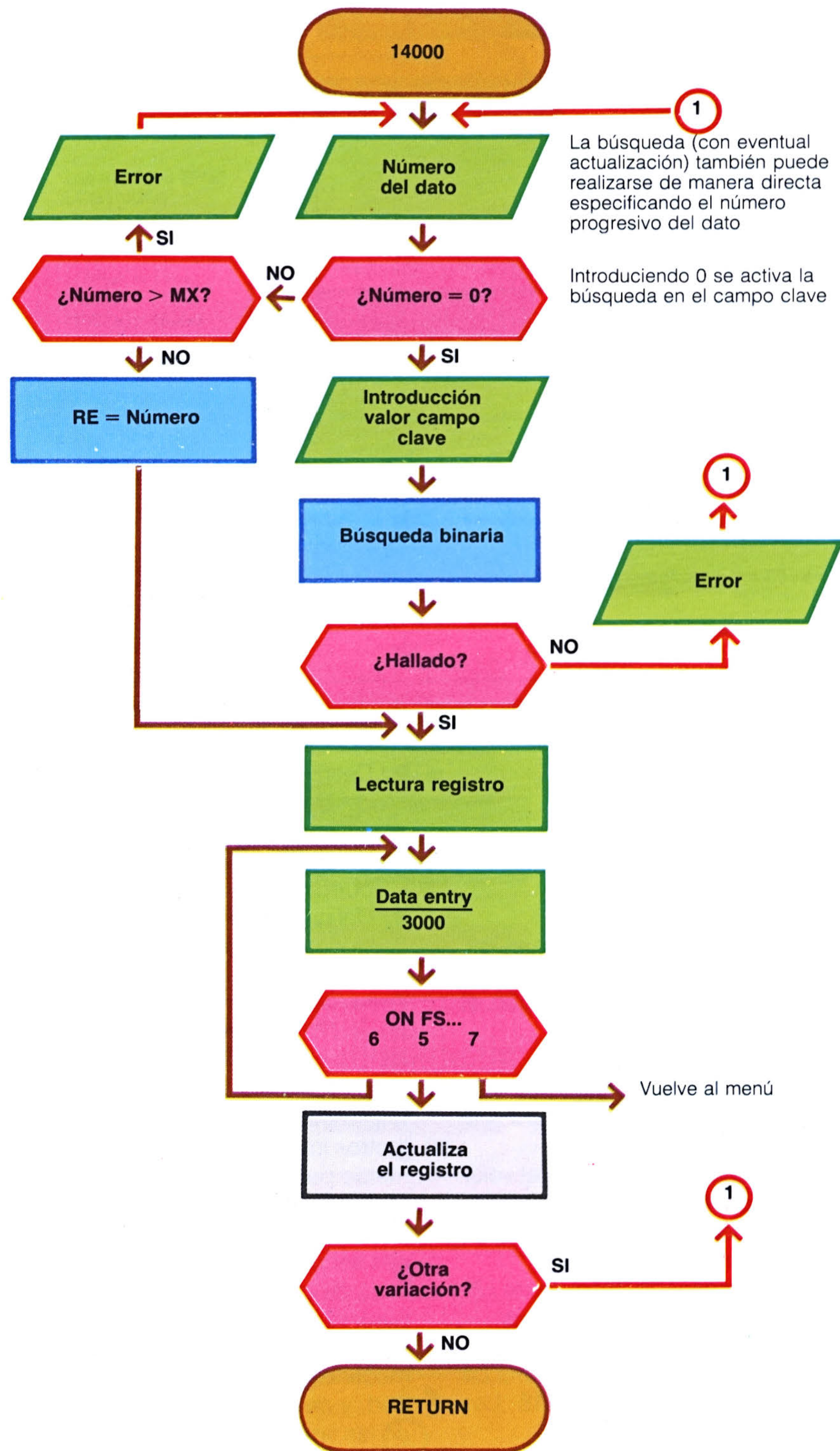
Según la primera hipótesis, el programa empieza por la petición del número del registro y pasa a la otra si el valor introducido es cero.
En la segunda hipótesis, búsqueda basada en el valor del campo clave, el fichero debe ordenarse previamente, puesto que la búsqueda puede proporcionar una solución negativa incluso si el valor existe.
Además, hay que rehacer el ordenado después de cada variación que haya interesado el campo clave.

Definición del formato de impresión. El diagrama de principio se muestra en las páginas 517 y 518. Las funciones que realiza son:

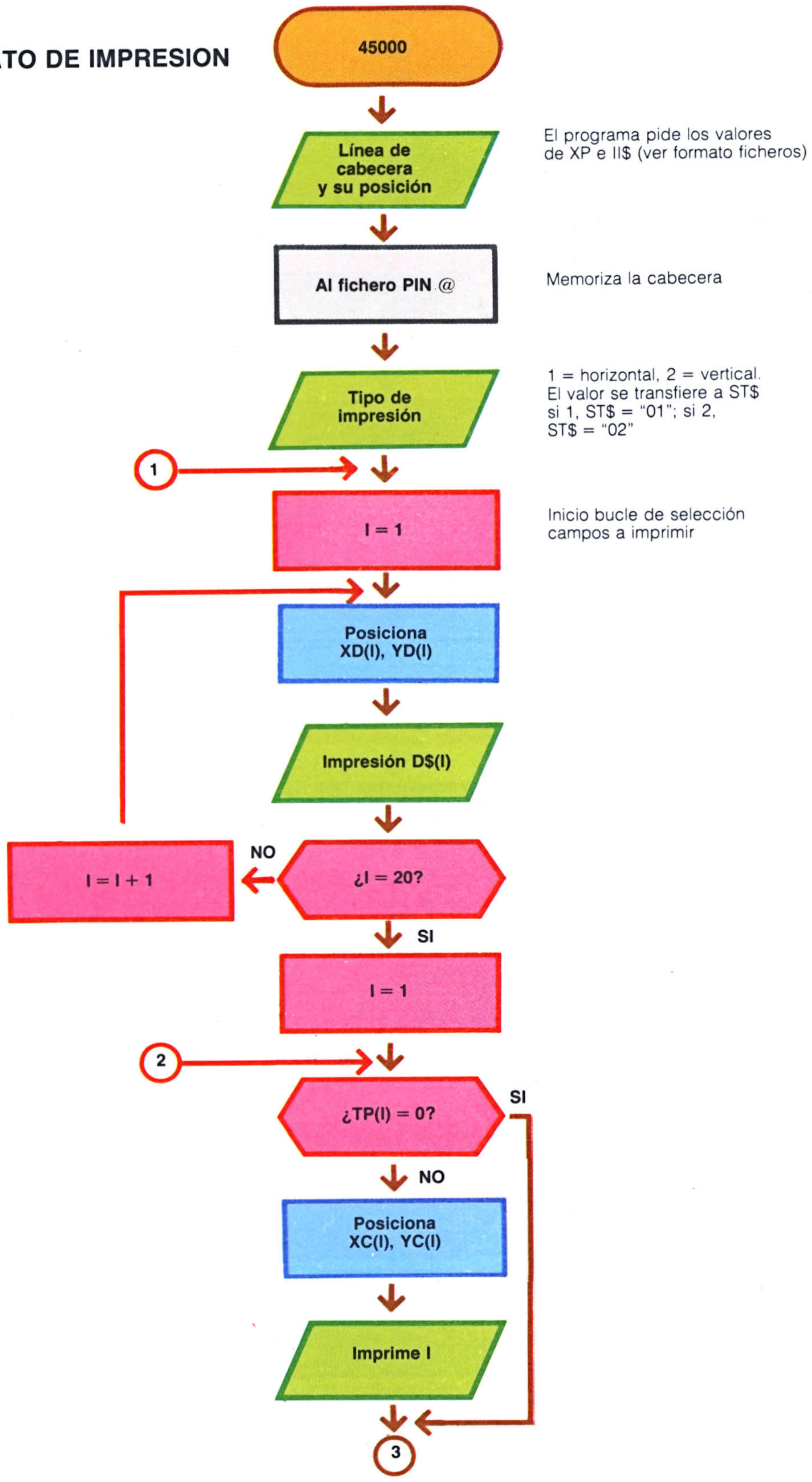
1 / Introducción de la línea de cabecera y control sobre su longitud
2 / Definición del tipo de impresión (horizontal, o sea con los datos correspondientes al mismo registro posicionados uno a continuación del otro en la misma línea, o bien vertical, o sea con los datos uno por cada línea)
3 / Definición de los campos a imprimir con la indicación de los que hay que totalizar

Todos los parámetros se introducen con un coloquio desarrollado con las instrucciones normales PRINT e INPUT y, por tanto, no ofrece ninguna dificultad.

Impresión. La subrutina de impresión utiliza los parámetros implantados en la fase de definición de los formatos para producir la salida en la forma deseada.
El diagrama representado en las páginas 2482 y 2483 sólo es válido como principio, porque algunas funciones secundarias, como por ejemplo la impresión de la línea de cabecera o el salto de página después de un número de líneas prefijado, no se han presentado.
El objeto del diagrama es principalmente mostrar el mecanismo de empleo de la matriz IS(*), índice del Sort, y de las funciones a realizar para la preparación de las líneas de impresión. El detalle del desarrollo de la subrutina depende mucho de la versión a que se refiere, por lo que deben consultarse los listados.


BUSQUEDA CON VARIACION
14000
1
Error
Número
del dato
La búsqueda (con eventual actualización) también puede realizarse de manera directa especificando el número progresivo del dato
SI
NO
¿Número > MX?
¿Número = 0?
Introduciendo 0 se activa la búsqueda en el campo clave
NO
SI
RE = Número
Introducción
valor campo
clave
1
Búsqueda binaria
Error
NO
¿Hallado?
SI
Lectura registro
Data entry
3000
ON FS...
6 5 7
Vuelve al menú
Actualiza
el registro
1
¿Otra
variación?
SI
NO
RETURN

DEFINICION DEL FORMATO DE IMPRESION



El programa pide los valores de XP e II$ (ver formato ficheros)

Memoriza la cabecera

1 = horizontal, 2 = vertical. El valor se transfiere a ST$ si 1, ST$ = "01"; si 2, ST$ = "02"

Inicio bucle de selección campos a imprimir

3

¿I = 20?

NO

I = I + 1

2

SI

Introducción campos a imprimir

Los campos se introducen mediante un input separado por un guión y precedido por una "T" si se desea la totalización (p.e., 2-T3-1)

¿Correcto?

NO

Error

1

SI

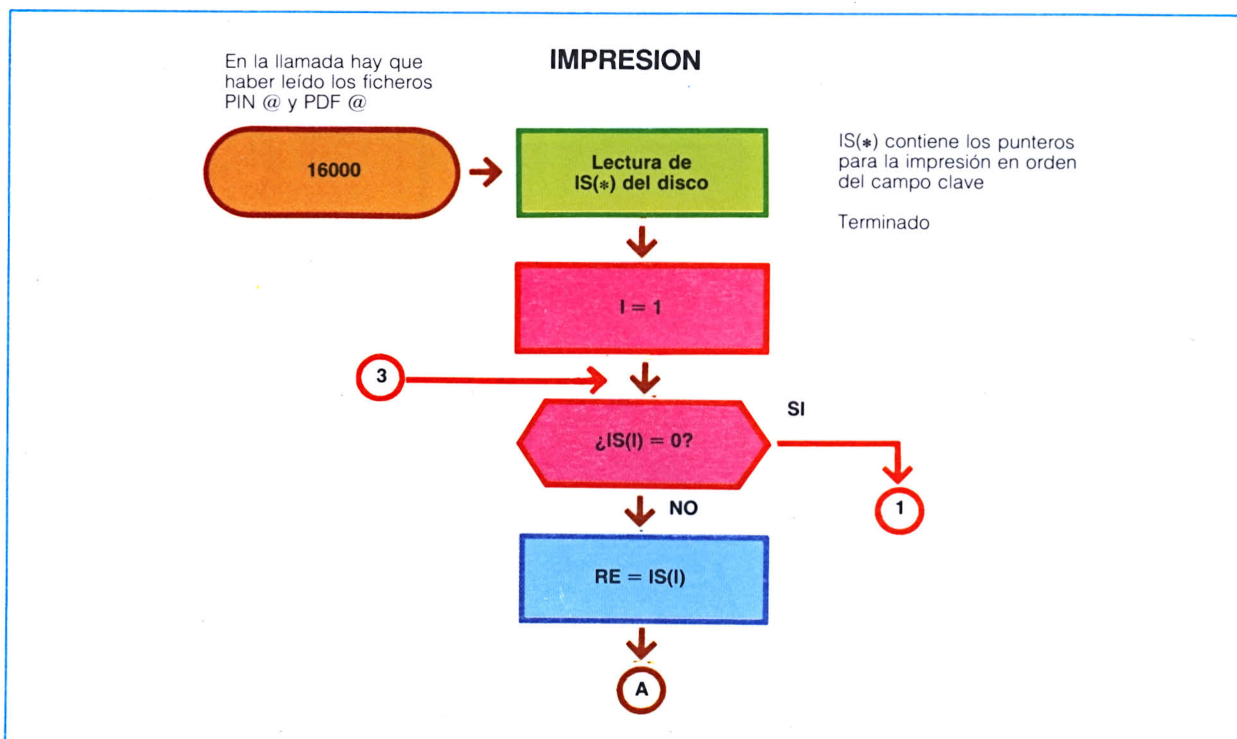
Transfiere los valores a NS$(*) y FT$(*)

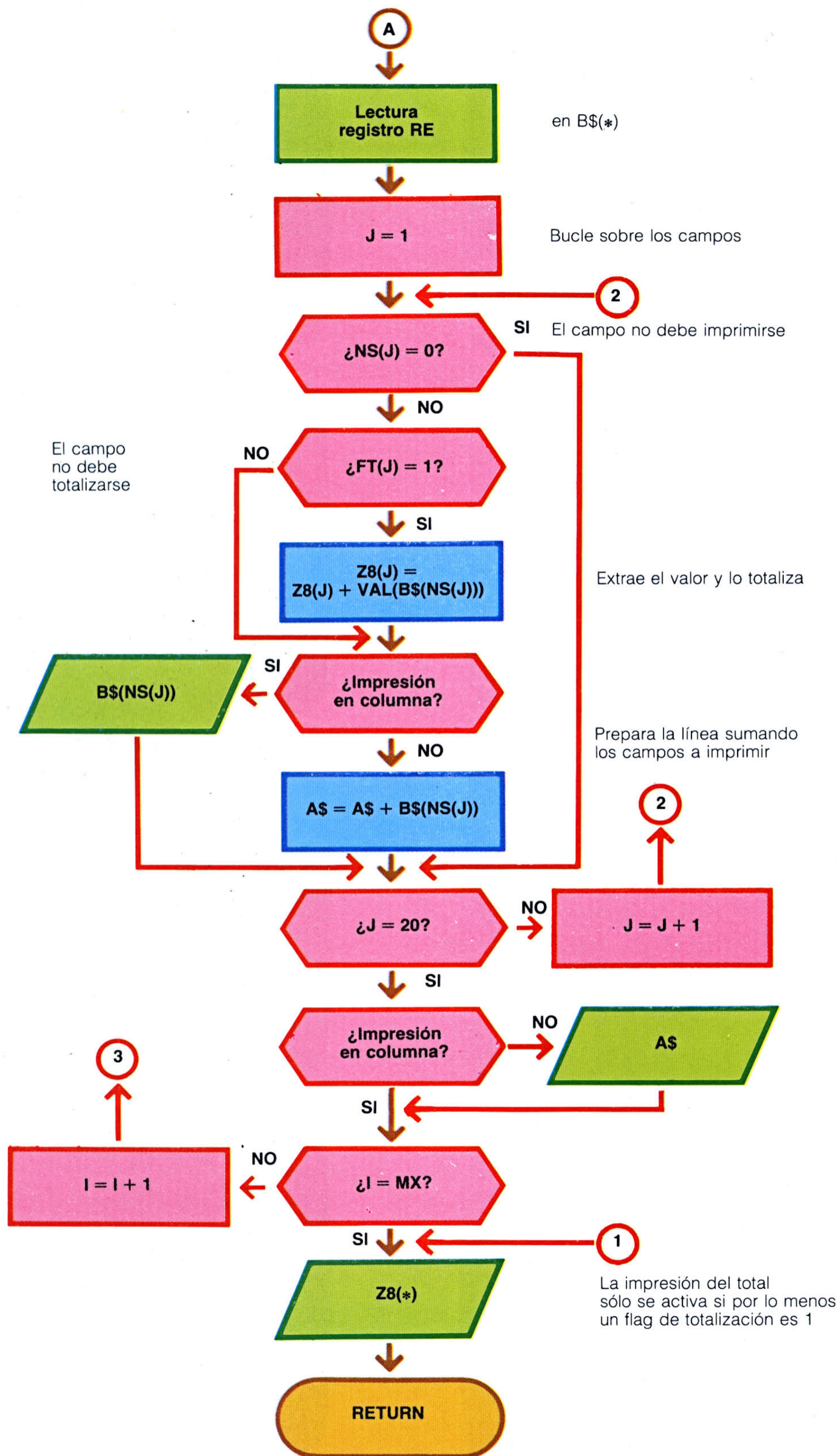
NS$(*) contiene los campos a imprimir en la secuencia deseada. FT$ contiene el flag de totalización (01 totaliza)

ST$ + FT$(1) + NS$(1)..... + FT$(20) + NS$(20)

Al fichero PDF @

RETURN

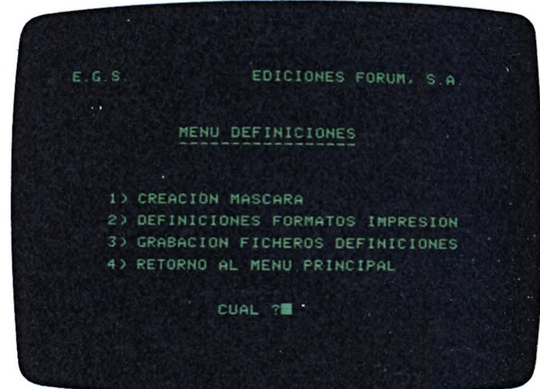



A
Lectura registro RE
en B$(*)
J = 1
Bucle sobre los campos
2
¿NS(J) = 0?
SI
El campo no debe imprimirse
NO
El campo no debe totalizarse
NO
¿FT(J) = 1?
SI
Z8(J) = Z8(J) + VAL(B$(NS(J)))
Extrae el valor y lo totaliza
B$(NS(J))
SI
¿Impresión en columna?
NO
Prepara la línea sumando los campos a imprimir
A$ = A$ + B$(NS(J))
2
¿J = 20?
NO
J = J + 1
SI
3
¿Impresión en columna?
NO
A$
SI
I = I + 1
NO
¿I = MX?
SI
1
Z8(*)
La impresión del total sólo se activa si por lo menos un flag de totalización es 1
RETURN

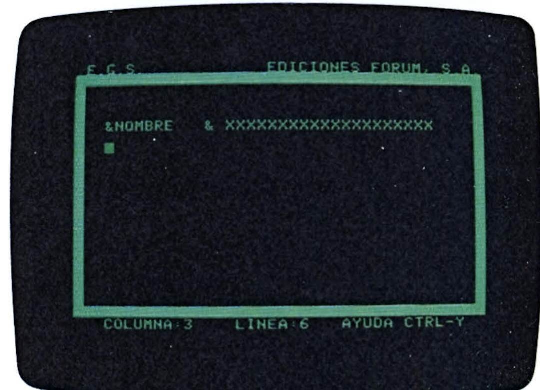
Base de Datos: menú de activación para el uso del programa (menú de trabajo). Las voces 1 y 5 piden el programa de Definición máscara y de Definición formatos impresión.



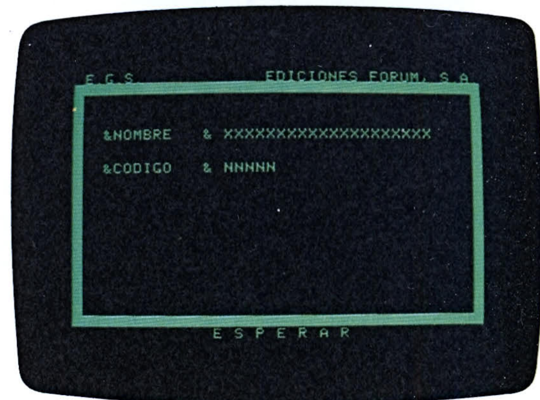
Menú definiciones: la primera función a activar es la Creación de la máscara (voz 1) que también define el formato del fichero. Los parámetros de definición de los campos, introducidos así, pueden memorizarse para un uso futuro con la función 3 (Grabación ficheros definiciones).



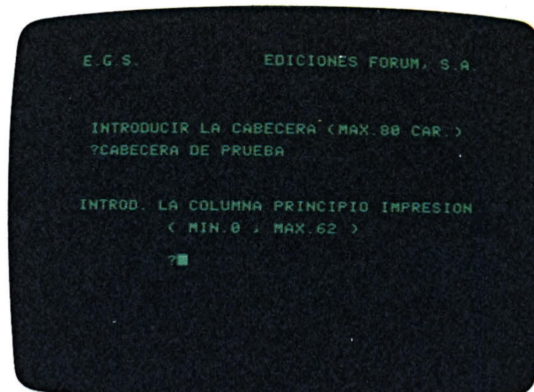
Ejemplo de máscara vídeo: los símbolos "&", como de costumbre, identifican la descripción asociada a cada campo. En la última línea de la pantalla aparece, constantemente actualizada, la posición actual del cursor.



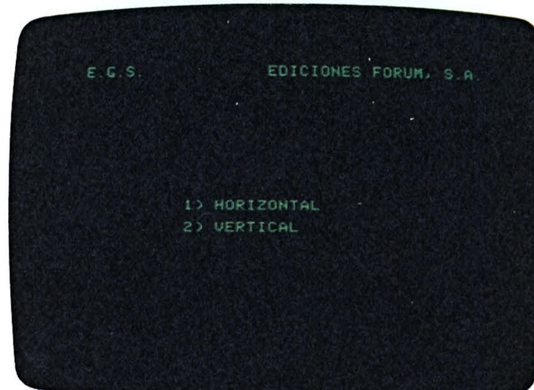
La misma máscara anterior en fase de creación.



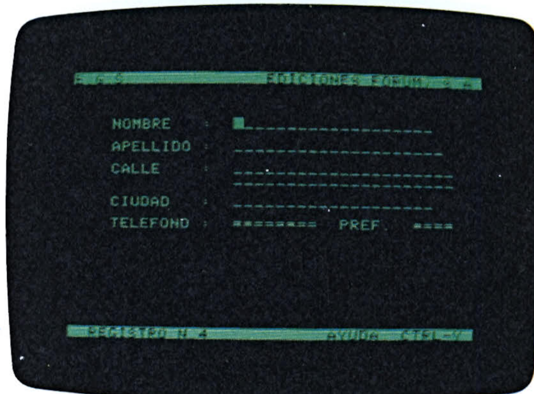
En la fase de impresión se ha previsto una línea de cabecera del tabulado: la foto muestra el coloquio de introducción. En la misma máscara aparece la petición del número de columna desde la cual debe empezar la impresión.



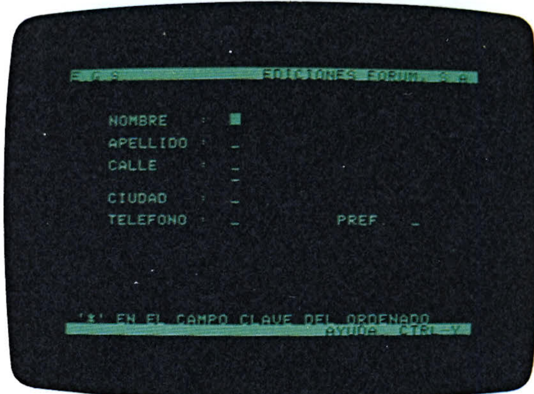
En la foto se muestra la petición del tipo de impresión:
1 / Horizontal: los datos de un registro se imprimen en la misma línea
2 / Vertical: cada campo ocupa una línea; por tanto, el registro se desarrolla verticalmente.



Ejemplo de introducción de datos en una máscara vídeo definida previamente.



Ejemplo de empleo de la función de ordenado: las definiciones de la máscara se han leído en el fichero de servicio y el programa espera la selección del campo a utilizar como clave.



Base de Datos: menú del programa de trabajo en la versión MSX.

```
E.G.S.                  Ediciones Forum, S.A.

               BASE DE DATOS

     « 1 »  definicion mascara
     « 2 »  introduccion datos
     « 3 »  ordenado
     « 4 »  busqueda con variacion
     « 5 »  definicion formatos impresion
     « 6 »  impresion
     « 0 »  Fin de tarea

                             CUAL ? «■ »
```

Menú del programa de creación de máscaras: también en esta versión, el empleo del procedimiento debe empezar por esta fase.

```
E.G.S.                  Editorial Forum, S.A.

              MENU DEFINICIONES

     1) creacion mascara
     2) definicion formatos impresion
     3) grabacion ficheros definiciones
     4) retorno al menu principal

                  CUAL ? ■
```

Creación de la máscara vídeo. Los símbolos utilizados son:
X = campo alfanumérico
N = campo numérico
C = campo resultado de un cálculo.

```
E.G.S.                  Editorial Forum, S.A.

 &CODIGO  & NNNNNNNN
 &DESCR.  & XXXXXXXXXXXXXXXXXXXX
            XXXXXXXXXXXXXXXXXXXX
 &ENTRADA & NNNNNNNN
 &SALIDA  & NNNNNNNN
 &EXISTENCIA& CCCCCCCCC
 ■

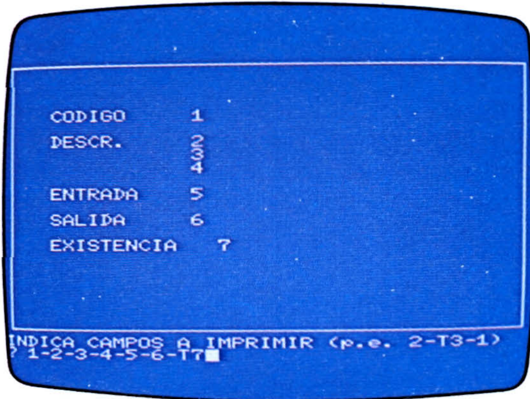
 Columna: 3   Linea: 16 F1 = AYUDA
```

Segunda fase de la definición de las máscaras: el programa espera la introducción de la fórmula para el cálculo del valor del campo definido de tipo "C".

```
 &CODIGO  &  1
 &DESCR.  & XXXXXXXXXXXXXXXXXXXX
            XXXXXXXXXXXXXXXXXXXX
 &ENTRADA &  5
 &SALIDA  &  6
 &EXISTENCIA&  7

INTROD.LA FORMULA DEL CAMPO 7 (desp.F5)
"5"-"6"■
```

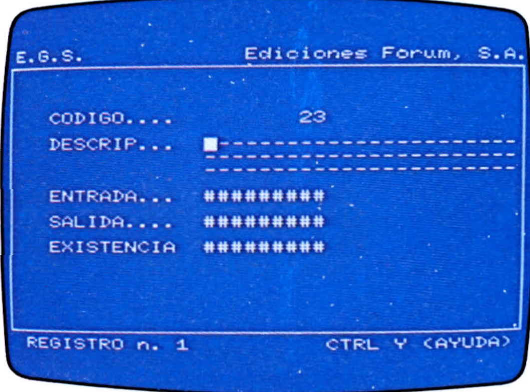
Definición de los formatos de impresión: el sistema presenta en pantalla todos los campos previstos en la máscara y espera la selección de los que hay que imprimir.



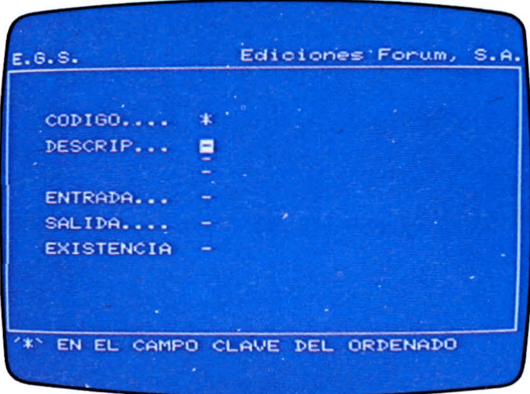
En la versión cinta, antes del acceso a la memoria de masa, el sistema emite los mensajes para las necesarias operaciones manuales de posicionado del soporte magnético.



Máscara de introducción de datos. Los parámetros asociados a la misma se han leído de la cinta después de su posicionado (ver foto anterior).



Ordenado. En esta foto se muestra la máscara correspondiente a esta función.



Extensiones y modificaciones

La subrutina de impresión ilustrada anteriormente genera la introducción del contenido del fichero, mientras que en muchas aplicaciones puede ser útil disponer de una salida selectiva que permita la presentación únicamente de los registros que tienen un determinado valor. Este tipo de impresión puede obtenerse modificando la subrutina anterior con las funciones:

- Indicación del campo a utilizar como elemento de control
- Introducción del valor
- Introducción de la rutina de selección con salto de la impresión si la comparación resulta negativa

Abajo se ha representado el diagrama de principio que muestra la manera de utilizar la subrutina anterior bajo esta nueva óptica. Naturalmente, la impresión debe modificarse para incluir la comprobación en el registro y la consiguiente decisión (en base al flag de respuesta).
En la misma figura (abajo) se ha representado la lógica de las implantaciones a incluir en la subrutina de impresión.
Obsérvese que esta subrutina no debe duplicarse para obtener los dos tipos de funcionamiento, sino que basta con forzar el flag al estado "ON" en caso de que no se pida ninguna selección.





Definiciones

```
10    REM    -------------
15    REM    DEFINICIONES
17    REM    BASE DE DATOS
20    REM    VERSION  DOS
30    REM    -------------
50    GOTO 40000
6000  REM
6010  PRINT D1$:
      PRINT D1$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2:
      PRINT D1$:
      RETURN
6200  REM
6210  PRINT D1$:
      PRINT D1$;"READ";NF$;",R";NR:
      INPUT AA$:
      PRINT D1$:
      RETURN
6400  REM
6410  PRINT D1$:
      PRINT D1$;"WRITE";NF$;",R";NR:
      PRINT  CHR$ (34) + AA$:
      PRINT D1$:
      RETURN
6600  REM
6610  PRINT D1$:
      PRINT D1$;"CLOSE";NF$:
      PRINT D1$:
      RETURN
40000 REM ------------------
40050 M3 = 20:
      D1$ =  CHR$ (4):
      B$ = "          ":
      FOR I = 1 TO 4:
        BA$ = BA$ + B$:
      NEXT I:
      B$ = "":
      DR$ = "1"
40052 H1$ = " E.G.S.             EDICION
      ES FORUM, S.A."
40055 DIM D$(20),TP(20),LC(20),ND(20),X
      D(20),YD(20),XC(20),YC(20),FO$(20
      ),FT$(20),NS$(20)
40200 REM ------------------
40230 GOSUB 46000
40240 ON OP GOSUB 41000,45000,42500,430
      00:
      GOTO 40200
41000 GOSUB 43500:
      HOME :
      X = 2:
      Y = 3:
      GOSUB 41150:
      GOSUB 41020:
      GOTO 41070
41020 A$ = " ":
      FOR I = 1 TO 40:
        L$ = L$ + A$:
      NEXT I:
      HTAB 1:
      VTAB 1:
      PRINT H1$;:
      INVERSE :
      PRINT L$;:
      FOR I = 1 TO 20:
        PRINT " ";:
        HTAB 40:
        PRINT " ";:
      NEXT I:
      PRINT L$;:
      L$ = "":
      NORMAL :
      RETURN
41070 REM ------------------
41080 FOR I = 1 TO 20:
        LC(I) = 0:
        TP(I) = 0:
        ND(I) = 0:
      NEXT I
41100 GOSUB 41310:
      A =  ASC (I$):
      IF A < 32 THEN 41180
41130 PRINT I$:
      X = X + 1:
      IF X > 39 THEN X = 39
41135 GOSUB 41150:
      GOTO 41100
41150 REM ------------------
41160 HTAB 4:
      VTAB 24:
      PRINT "COLUMNA:
      ";X - 1; TAB( 17);"LINEA:
      ";Y - 2; TAB( 27);"AYUDA CTRL-Y";
      :
      RETURN
41180 REM
41210 IF A = 19 THEN X = X + 1:
      IF X > 39 THEN X = 2:
      A = 26:
      GOTO 41240
41220 IF A = 1 THEN X = X - 1:
      IF X < 2 THEN X = 39:
      A = 23
41230 IF A = 23 THEN Y = Y - 1:
      IF Y < 3 THEN Y = 3
41240 IF A = 26 THEN Y = Y + 1:
      IF Y > 22 THEN Y = 22
41250 IF A = 13 THEN X = 2:
      Y = Y + 1:
      IF Y > 22 THEN Y = 22
41260 IF A = 9 THEN  GOSUB 41360
41270 IF A = 4 THEN  GOSUB 41400
41280 IF A = 5 THEN 41460
41290 IF A = 25 THEN  GOSUB 44000
41300 GOSUB 41150:
      GOTO 41100
41310 REM -------------------------
41320 HTAB X:
      VTAB Y:
      GET I$:
      RETURN
41360 REM -------------------------
41370 P1 = 0:
      P2 = X - 1:
      P3 = 37:
      FOR I = P3 TO P2 STEP  - 1:
        HTAB 39 - P1:
        VTAB Y:
        PRINT  CHR$ ( SCRN( I,2
        * (Y - 1)) + 16 *  SCRN( I,2
        * (Y - 1) + 1));:
        P1 = P1 + 1:
      NEXT :
      HTAB X:
      VTAB Y:
      PRINT " ";:
      RETURN
41400 REM
41430 P1 = 0:
      P2 = X:
      P3 = 38:
      FOR I = P2 TO P3:
        HTAB X + P1:
        VTAB Y:
        PRINT  CHR$ ( SCRN( I,2
        * (Y - 1)) + 16 *  SCRN( I,2
        * (Y - 1) + 1));:
        P1 = P1 + 1:
      NEXT :
      PRINT " ";:
      RETURN
41460 REM -------------------------
41490 HTAB 1:
      VTAB 24:
      PRINT  SPC( 14);"E S P E R A R";
       SPC( 10);" ";:
      YY = 2:
      PD = 0:
      PC = 0:
      FOR JJ = 2 TO 21:
        J = JJ * 40:
        YY = YY + 1:
        FOR K = 1 TO 38:
          XX = XX + 1:
          A$ =  CHR$ ( SCRN( K,2
          * JJ) + 16 *  SCRN( K,2
          * JJ + 1)):
          A$ =  CHR$ ( ASC (A$)
          - 128):
          IF A$ = "&" THEN 41660
41550     IF A$ = "X" OR A$ = "N"
          OR A$ = "C" THEN 41570
41560     IF A$ < > " " THEN 41770
41565     GOTO 41730
41570     PC = PC + 1:
          IF PC > M3 THEN 41810
41580     IF A$ = "X" THEN TP(PC)
          = 3
41582     IF A$ = "N" THEN TP(PC)
          = 1
41584     IF A$ = "C" THEN TP(PC)
          = 2
41590     LC(PC) = 1:
          YC(PC) = YY:
          XC(PC) = K + 1:
          ND(PC) = 0:
          W6 = 0
41600     K = K + 1:
          IF K = 40 THEN 41770
41610     IF W6 = 1 AND A$ = "X"
          THEN 41770
41620     AA$ =  CHR$ ( SCRN( K,2
          * JJ) + 16 *  SCRN( K,2
          * JJ + 1)):
          AA$ =  CHR$ ( ASC (AA$)
          - 128):
          IF AA$ < > A$ THEN 41630
41624     LC(PC) = LC(PC) + 1:
          IF W6 = 1 THEN ND(PC)
          = ND(PC) + 1
41626     GOTO 41600
41630     V =  SCRN( K,2 * JJ)
          + 16 *  SCRN( K,2 * JJ
          + 1):
          V = V - 128:
          IF V = 46 AND W6 = 0
          THEN LC(PC) = LC(PC)
          + 1:
          W6 = 1:
          GOTO 41600
41640     IF V = 32 OR V = 127
          THEN K = K - 1:
          GOTO 41730
```

```
41650      IF V <  > 38 THEN 41770
41660      REM
41670      PD = PD + 1:
           IF PD > M3 THEN 41810
41680      D$(PD) = "":
           XD(PD) = K + 2:
           YD(PD) = YY
41690      K = K + 1:
           IF K = 40 THEN 41770
41700      AA$ =  CHR$ ( SCRN( K,2
            * JJ) + 16 *  SCRN( K,2
            * JJ + 1)):
           AA$ =  CHR$ ( ASC (AA$)
            - 128):
           IF AA$ = "&" THEN 41730
41710      D$(PD) = D$(PD) + AA$:
           GOTO 41690
41730 NEXT K,JJ:
      AA = 0:
      FOR I = 1 TO M3:
         AA = AA + LC(I):
      NEXT I:
      IF AA > 128 THEN T$ = "LONG. CAMP
      OS > 128":
      GOTO 41820
41760 GOTO 41850
41770 REM ----------------------
41800 T$ = "ERROR EN LA LINEA " +
       STR$ (YY - 2):
      GOTO 41820
41810 T$ = "CAMPOS O DESCRIP.> "
       +  STR$ (M3)
41820 HTAB 2:
      VTAB 24:
      PRINT T$;" :
      CORREGIR  ";BP$;:
      FOR I = 1 TO 4000:
      NEXT I:
      HTAB 1:
      VTAB 24:
      PRINT  SPC( 39);:
      X = 2:
      Y = YY:
      GOSUB 41150:
      GOTO 41070
41850 REM ----------------------
41875 IF PC = 0 THEN  RETURN
41880 MX = PC:
      NC = 0:
      FOR I = 1 TO MX:
         IF TP(I) = 3 THEN 41930
41910    HTAB XC(I):
         VTAB YC(I):
         PRINT  SPC( LC(I)):
         HTAB XC(I):
         VTAB YC(I):
         PRINT  STR$ (I);
41930 NEXT I:
      HTAB 1:
      VTAB 24:
      PRINT  SPC( 40):
      FOR I = 1 TO MX:
         FO$(I) = "":
         IF TP(I) <  > 2 THEN 42070
41970    HTAB 1:
         VTAB 20:
         PRINT "FORMULA DEL CAMPO ";I;"
         ? (DESPUES CTRL E)";:
         Y = 24:
         X = 1:
         HTAB 1:
         VTAB 24:
         PRINT  SPC( 39);BP$;
41990    GOSUB 41310:
         IF I$ =  CHR$ (1) THEN X
          = X - 1:
         IF X < 1 THEN X = 1
42000    IF I$ =  CHR$ (19) THEN X
          = X + 1:
         IF X > 31 THEN X = 31
42010    IF I$ =  CHR$ (5) THEN 42050
42020    IF  ASC (I$) < 32 THEN 41990
42030    PRINT I$;:
         X = X + 1:
         IF X > 31 THEN X = 31
42040    GOTO 41990
42050    Y = 23:
         FOR H = 0 TO 30:
            V =  SCRN( H,2 * Y) + 16
             *  SCRN( H,2 * Y + 1):
            V = V - 128:
            FO$(I) = FO$(I) +  CHR$ (V)
            :
         NEXT H
42070 NEXT I:
      FOR I = 1 TO 1000:
      NEXT :
      W5 = 2:
      RETURN
42500 REM ----------------------
42530 HOME :
      HTAB 7:
      VTAB 9:
      PRINT "INTRODUCIR EL DISCO DATOS"
      :
      HTAB 8:
      VTAB 20:
      PRINT "DESPUES PULSAR UNA TECLA";
      :
      GET I$:
      HOME :
      HTAB 14:
      VTAB 12:
      PRINT "E S P E R A R":
      NF$ = "DEFC":
      LR = 26:
      GOSUB 6000:
      FOR I = 1 TO M3:
         L =  LEN (D$(I)):
         IF L < 10 THEN D$(I) = D$(I)
          +  LEFT$ (BA$,10 - L):
         GOTO 42600
42590    D$(I) =  LEFT$ (D$(I),10)
42600    A$ = "0" +  STR$ (XD(I)):
         R$ =  RIGHT$ (A$,2):
         A$ = "0" +  STR$ (YD(I)):
         R$ = R$ +  RIGHT$ (A$,2):
         R$ = R$ + D$(I):
         A$ = "0" +  STR$ (XC(I)):
         R$ = R$ +  RIGHT$ (A$,2):
         A$ = "0" +  STR$ (YC(I)):
         R$ = R$ +  RIGHT$ (A$,2):
         A$ = "0" +  STR$ (LC(I)):
         R$ = R$ +  RIGHT$ (A$,2):
         A$ = "0" +  STR$ (TP(I)):
         R$ = R$ +  RIGHT$ (A$,2):
         A$ = "0" +  STR$ (ND(I)):
         R$ = R$ +  RIGHT$ (A$,2)
42680    NR = I:
         AA$ = R$:
         GOSUB 6400:
      NEXT I:
      GOSUB 6600:
      NF$ = "FORC":
      LR = 34:
      GOSUB 6000:
      FOR I = 1 TO M3:
         Z$ =  STR$ (I):
         IF I < 10 THEN Z$ = "0"
          + Z$
42730    L =  LEN (FO$(I)):
         IF L < 30 THEN FO$(I) = FO$(I)
          +  LEFT$ (BA$,30 - L):
         GOTO 42750
42740    FO$(I) =  LEFT$ (FO$(I),30)
42750    AA$ = Z$ + FO$(I) + "  ":
         NR = I:
         GOSUB 6400:
      NEXT I:
      GOSUB 6600:
      NF$ = "PINC":
      LR = 82:
      GOSUB 6000:
      AA$ = II$:
      NR = 1:
      GOSUB 6400:
      GOSUB 6600:
      NF$ = "PDFC":
      LR = 82:
      GOSUB 6000:
      FOR I = 1 TO M3:
         ST$ = ST$ + FT$(I) + NS$(I):
      NEXT I:
      NR = 1:
      AA$ = ST$:
      GOSUB 6400:
      GOSUB 6600
42830 RETURN
43000 REM ----------------------
43010 HOME :
      HTAB 8:
      VTAB 10:
      PRINT "INTRODUCIR EL DISCO PROGRA
      MAS":
      HTAB 8:
      VTAB 20:
      PRINT "DESPUES PULSAR UNA TECLA":
      GET I$:
      PRINT D1$:
      PRINT D1$;"RUN MAIN.FUENTE,D1"
43500 RETURN
44000 REM ----------------------
44030 PRINT BP$;:
      HTAB 1:
      VTAB 24:
      PRINT "BARRA ESPACIADORA PARA CON
      TINUAR          ";:
      GOSUB 44180:
      HTAB 1:
      VTAB 24:
      PRINT "CTRL.I:
       INSERCION DE UN CARACTER       ";
      :
      GOSUB 44180:
      HTAB 1:
      VTAB 24:
      PRINT "CTRL.D:
       BORRADO DE UN CARACTER         ";
      :
      GOSUB 44180:
      HTAB 1:
      VTAB 24:
      PRINT "RETURN:
       VA AL PRINCIPIO                ";
      :
44110 GOSUB 44180:
      HTAB 1:
      VTAB 24:
      PRINT "CTRL.E:
       CONTROLA,CONVALIDA Y GRABA   ";:
      GOSUB 44180:
      HTAB 1:
      VTAB 24:
      PRINT "CTRL.A,S,Z,W:
       MUEVEN EN LAS 4 DIRECCIO.";:
      GOSUB 44180:
      HTAB 1:
      VTAB 24:
      PRINT  SPC( 39);:
      RETURN
44180 GET J$:
      IF J$ = " " THEN  RETURN
44190 GOTO 44180
44500 REM ----------------------
44530 HOME :
      HTAB 10:
      VTAB 8:
```

```
PRINT "INTRODUCIR EL DISCO DATOS"
:
HTAB 10:
VTAB 20:
PRINT "DESPUES PULSAR UNA TECLA "
;:
GET I$:
HOME :
HTAB 14:
VTAB 12:
PRINT "E S P E R A R":
NF$ = "DEFC":
      LR = 26:
      GOSUB 6000:
      FOR I = 1 TO M3:
         NR = I:
         GOSUB 6200:
         XD(I) =  VAL ( MID$ (AA$,1,2))
         :
         YD(I) =  VAL ( MID$ (AA$,3,2))
         :
         D$(I) =  MID$ (AA$,5,10)
44610    XC(I) =  VAL ( MID$ (AA$,15,2)
         ):
         YC(I) =  VAL ( MID$ (AA$,17,2)
         ):
         LC(I) =  VAL ( MID$ (AA$,19,2)
         ):
         TP(I) =  VAL ( MID$ (AA$,21,2)
         ):
         ND(I) =  VAL ( MID$ (AA$,23,2)
         ):
      NEXT I:
      GOSUB 6600:
      NF$ = "FORC":
      LR = 34:
      GOSUB 6000:
      FOR I = 1 TO M3:
         NR = I:
         GOSUB 6200
44680    FO$(I) =  MID$ (AA$,3,30):
      NEXT I:
      GOSUB 6600:
      RETURN
45000 REM ------------------------
45025 IF W5 = 0 THEN W5 = 2:
      GOSUB 44500
45030 HOME :
      HTAB 1:
      VTAB 1:
      PRINT H1$;:
      HTAB 3:
      VTAB 7:
      PRINT "INTRODUCIR LA CABECERA (MA
      X.80 CAR.)";:
      HTAB 3:
      VTAB 9:
      INPUT I$:
      A =  LEN (I$):
      IF A > 80 THEN 45030
45050 IF A = 80 THEN XP = 0:
      GOTO 45090
45060 HTAB 2:
      VTAB 14:
      PRINT "INTROD. LA COLUMNA PRINCIP
      IO IMPRESION":
      HTAB 11:
      VTAB 16:
      PRINT "( MIN.0 ; MAX.";80 - A;" )
      ";
      HTAB 11:
      VTAB 19:
      INPUT XP:
      IF XP < 0 OR XP > 80 THEN 45030
45090 IF XP < 10 THEN A$ = "0" +
       STR$ (XP):
      GOTO 45100
45095 A$ =  STR$ (XP)
45100 II$ = A$ + I$ + MID$ (BA$,1,80
      -  LEN (I$)):
      HOME :
      HTAB 1:
      VTAB 1:
      PRINT H1$;:
      HTAB 12:
      VTAB 13:
      PRINT "1) HORIZONTAL":
      HTAB 12:
      VTAB 15:
      PRINT "2) VERTICAL"
45170 VTAB 17:
      HTAB 23:
      GET A$:
      IF A$ <  > "1" AND A$ <  > "2"
       THEN 45170
45180 ST$ = "0" + A$:
      HOME :
      HTAB 1:
      VTAB 1:
      PRINT H1$;:
      HTAB 3:
      VTAB 6:
      PRINT "EN LA SIGUIENTE MASCARA DE
      BERAN":
      PRINT :
      HTAB 3:
      PRINT "SELECCIONARSE LOS CAMPOS A
       IMPRIMIR,":
      HTAB 3:
      VTAB 10:
      PRINT "INSERTANDO LA LETRA 'T' SI
       EL CAMPO":
      PRINT :
      HTAB 3:
      PRINT "(NUMERICO O DE CALCULO) DE
      BE":
      PRINT :
      HTAB 3:
      PRINT "TOTALIZARSE."
45220 HTAB 1:
      VTAB 18:
      PRINT "ES.:
       2-T1-15-T6 =IMPRESION DE LOS CAM
      POS 2,1,15,6 CON TOTALIZACION DE
      LOS CAMPOS 1 Y 6":
      HTAB 11:
      VTAB 24:
      PRINT "PULSAR UNA TECLA";:
      GET J$
45250 HOME :
      GOSUB 41020:
      FOR I = 1 TO M3:
         IF  LEFT$ (D$(I),3) = "   "
          OR  LEFT$ (D$(I),1) = ""
          THEN 45270
45260    HTAB XD(I):
         VTAB YD(I):
         PRINT D$(I)
45270 NEXT I:
      FOR I = 1 TO M3:
         IF TP(I) = 0 THEN 45300
45290    HTAB XC(I):
         VTAB YC(I):
         PRINT  STR$ (I)
45300 NEXT I:
      HTAB 1:
      VTAB 24:
      PRINT "INDICA LOS CAMPOS A IMPRIM
      IR (ES.2-T3-1)";:
      HTAB 1:
      VTAB 24:
      INPUT B$:
      LL =  LEN (B$):
      A$ = "":
      C = 0:
      FOR I = 1 TO LL:
         C$ =  MID$ (B$,I,1):
         IF C$ = "-" THEN  GOSUB 45370:
         GOTO 45360
45350    A$ = A$ + C$
45360 NEXT I:
      I = I - 1
45370 REM ------------------------
45380 C = C + 1:
      L =  LEN (A$):
      IF  LEFT$ (A$,1) = "T" THEN FT$(C
      ) = "01":
      L = L - 1:
      A$ =  MID$ (A$,2,L):
      GOTO 45400
45395 FT$(C) = "00"
45400 V =  VAL (A$):
      IF V = 0 THEN 45470
45410 IF TP(V) = 0 THEN 45470
45420 IF TP(V) = 3 THEN FT$(C) = "00"
45430 IF L > 2 OR L < 1 THEN 45470
45440 IF L = 1 THEN NS$(C) = "0"
       + A$:
      GOTO 45450
45445 NS$(C) = A$
45450 IF I = LL THEN 45560
45460 A$ = "":
      RETURN
45470 REM ------------------------
45480 HOME :
      HTAB 14:
      VTAB 14:
      PRINT "E R R O R";BP$;:
      FOR I = 1 TO 1200:
      NEXT :
      FOR I = 1 TO M3:
         FT$(I) = "":
         NS$(I) = "":
      NEXT :
      HOME :
      HTAB 11:
      VTAB 11:
      PRINT "1) CONTINUO":
      HTAB 11:
      VTAB 13:
      PRINT "2) VUELVO AL MENU":
      HTAB 14:
      VTAB 18:
      PRINT "CUAL ?";
45530 GET A$:
      IF A$ <  > "1" AND A$ <  > "2"
       THEN 45530
45540 IF A$ = "2" THEN  RETURN
45550 GOTO 45250
45560 FOR I = C + 1 TO M3:
         FT$(I) = "00":
         NS$(I) = "00":
      NEXT :
      IF ST$ = "02" THEN 45640
45600 B = 0:
      FOR I = 1 TO M3:
         A =  VAL (NS$(I)):
         IF A = 0 THEN 45620
45610    B = B + LC(A)
45620 NEXT I:
      IF B > 80 THEN 45470
45640 RETURN
46000 REM ------------------------
46030 HOME :
      HTAB 1:
      VTAB 1:
      PRINT H1$;:
      HTAB 13:
      VTAB 6:
      PRINT "MENU DEFINICIONES":
      HTAB 13:
      VTAB 7:
      PRINT "-----------------":
      HTAB 6:
      VTAB 12:
      PRINT "1) CREACION MASCARA"
46070 HTAB 6:
```

```
       VTAB 14:
       PRINT "2) DEFINICIONES FORMATOS I
       MPRESION":
       HTAB 6:
       VTAB 16:
       PRINT "3) GRABACION FICHEROS DEFI
       NICIONES":
       HTAB 6:
       VTAB 18:
       PRINT "4) RETORNO AL MENU PRINCIP
       AL":
       HTAB 17:
       VTAB 22:
       PRINT "CUAL ?";
46110  GET A$:
       IF A$ < "1" OR A$ > "4" THEN 4611
       0
46120  PRINT A$;:
       FOR I = 1 TO 1000:
       NEXT :
       OP = VAL (A$):
       RETURN
```

Main

```
10     REM   -------------
20     REM    BASE DE DATOS
25     REM     VERSION DOS
30     REM   -------------
50     M3 = 20:
       MR = 120
60     DIM TP(M3),LC(M3),ND(M3),XC(M3),Y
       C(M3),XD(M3),YD(M3),HT(M3),HL(M3)
       ,HN(M3),IS(MR),NS(M3),FT(M3),Z8(M
       3),B$(M3),D$(M3),F$(M3),DA$(MR)
70     RESTORE :
       READ NV:
       FOR I = 1 TO NV:
          READ A$:
       NEXT
80     FOR I = 1 TO 5:
          READ OP$(I):
       NEXT
90     FOR I = 1 TO 6:
          READ FZ$(I):
       NEXT
220    REM   --- MAIN ---
230    GOSUB 1300
240    IF OP = NV THEN  TEXT :
       HOME :
       END
250    ON OP GOSUB 12000,4500,5500,14000
       ,12000,16000
290    GOTO 230
500    REM
530    TEXT :
       HOME :
       BP$ =  CHR$ (7):
       D$ =  CHR$ (4):
       DR$ = "1":
       B$ = "          ":
       BA$ = B$ + B$ + B$ + B$:
       B$ = "__________":
       BU$ = B$ + B$ + B$ + B$:
       B$ = "==========":
       BE$ = B$ + B$ + B$ + B$:
       B$ = "00000000000000000000":
       ZE$ = B$ + B$ + B$ + B$
540    INVERSE :
       PRINT "E.G.S.              EDICION
       ES FORUM, S.A.";:
       NORMAL
565    HE$ = "CTRL E. INTRODUCCION - CTR
       L R. VUELVE AL MENU - CTRL A,S,Z,
       W. MUEVEN EN LAS CUATRO DIRECCION
       ES - CTRL I. INSERCION - CTRL D.
       BORRADO - ESC . ANULACION - PULSA
       R UNA TECLA - "
580    RETURN
600    REM  --- ESPERAR ---
610    GOSUB 500:
       HTAB 14:
       VTAB 13:
       PRINT "E S P E R A R":
       RETURN
1000   REM
1010   X1 = X - 1:
       Y1 = Y - 1:
       CH =  SCRN( X1,2 * Y1) + 16
        *  SCRN( X1,2 * Y1 + 1):
       HTAB X:
       VTAB Y:
       FLASH :
       IF FC = 0 THEN  INVERSE
1040   PRINT  CHR$ (CH);:
       HTAB X:
       NORMAL :
       RETURN
1100   REM
1110   X = X1 + 1:
       Y = Y1 + 1:
       NORMAL :
       HTAB X:
       VTAB Y:
       PRINT  CHR$ (CH);:
       HTAB X:
       RETURN
1200   REM
1210   A0 =  PEEK ( - 16384):
       IF A0 > 126 THEN  POKE  - 16368,0
       :
       A0 = A0 - 128:
       RETURN
1220   GOTO 1210
1300   REM
1330   GOSUB 500:
       RESTORE :
       READ NV:
       IF NV = 0 THEN  RETURN
1350   L = 0:
       FOR I = 1 TO NV:
          READ VO$(I):
          LL =  LEN (VO$(I)):
          IF LL > L THEN L = LL
1370   NEXT I:
       XI = (40 - L) / 2:
       YI = ((24 - NV * 2) / 2) + 1:
       X = XI - 2:
       Y = YI:
       FOR I = 1 TO NV:
          HTAB X:
          VTAB Y:
          PRINT I;" - ";VO$(I):
          Y = Y + 2:
       NEXT I:
       Y = Y + 1
1410   HTAB X:
       VTAB Y:
       PRINT "CUAL ?";:
       GOSUB 1200:
       OP$ =  CHR$ (A0):
       OP =  VAL (OP$):
       IF OP < 1 OR OP > NV THEN
        PRINT BP$;:
       GOTO 1410
1440   FLASH :
       PRINT OP:
       NORMAL :
       FOR I = 1 TO 1000:
       NEXT I:
       RETURN
2000   REM  -----------------------
2020   NC = M3
2030   GOSUB 500:
       HTAB 1:
       VTAB 24:
       INVERSE :
       PRINT  SPC( 39);:
       NORMAL
2032   FOR I = 1 TO NC:
          IF D$(I) =  LEFT$ (BA$,10)
           THEN 2037
2035      HTAB XD(I):
          VTAB YD(I):
          PRINT D$(I);
2037   NEXT
2038   FOR IC = 1 TO NC:
          IF TP(IC) = 0 THEN 2170
2039      IF TP(IC) = 4 THEN A$ = B$(IC)
          :
          GOTO 2160
2040      B$(IC) =  LEFT$ (B$(IC),LC(IC)
          ):
          L =  LEN (B$(IC)):
          IF L < LC(IC) THEN B$(IC)
           = B$(IC) +  LEFT$ (BA$,LC(IC)
           - L)
2080      IF TP(IC) <  > 1 AND TP(IC)
           <  > 2 THEN 2120
2085      IF ND(IC) = 0 THEN 2112
2090      PD = ND(IC):
          C1 = LC(IC) - PD - 1:
          C2 = LC(IC) - C1 - 1:
          B$(IC) =  LEFT$ (BA$,C1)
           + "." +  LEFT$ (BA$,C2):
          A$ =  LEFT$ (BE$,C1) + "."
           +  LEFT$ (BE$,C2):
          GOTO 2160
2112      B$(IC) =  LEFT$ (BA$,LC(IC)):
          A$ =  LEFT$ (BE$,LC(IC)):
          GOTO 2160
2120      B$(IC) =  LEFT$ (BA$,LC(IC)):
          Z$ = "":
          L = 0:
          GOTO 2150
2150      A$ = Z$ +  MID$ (BU$,1,LC(IC)
           - L)
2160      HTAB XC(IC):
          VTAB YC(IC):
          PRINT A$
2170   NEXT IC:
       RETURN
2500   REM
2530   L =  LEN (Z$):
       IF L = 0 THEN  RETURN
2535   FOR I = 1 TO L:
          C =  ASC ( MID$ (Z$,I,1)):
          IF C <  > 32 THEN I1 = I:
          GOTO 2560
2550   NEXT I:
       Z$ = "":
       RETURN
2560   FOR I = L TO 1 STEP  - 1:
          C =  ASC ( MID$ (Z$,I,1)):
          IF C <  > 32 THEN I2 = I:
          GOTO 2590
2580   NEXT I
2590   KO = I2 - I1 + 1:
       Z$ =  MID$ (Z$,I1,KO):
       RETURN
2700   REM
2730   XT = (42 -  LEN (TI$)) / 2:
       HTAB XT:
       VTAB YT:
       PRINT TI$;:
       RETURN
2800   REM
2830   A$ =  LEFT$ (BA$,38):
       HTAB 2:
       VTAB YT:
       PRINT A$;:
       RETURN
2900   REM
2930   Z2$ = "":
       A =  LEN (Z$):
       I = 1
```

```
2940  Z1$ =  MID$ (Z$,I,1):
      IF Z1$ <  > " " THEN Z2$ = Z2$
       + Z1$
2945  I = I + 1:
      IF I <  = (A) THEN 2940
2950  Z$ = Z2$:
      RETURN
3000  REM
3030  CC = 1:
      FC = 1:
      INVERSE :
      HTAB 26:
      VTAB 24:
      PRINT "AYUDA. CTRL-Y";:
      NORMAL
3032  IF TP(CC) <  > 2 THEN 3042
3035  GOTO 8500
3037  GOTO 3080
3040  REM
3042  IF TP(CC) = 4 OR TP(CC) = 0
       THEN 3080
3045  XC = XC(CC):
      YC = YC(CC):
      LO = LC(CC):
      GOSUB 3500:
      IF AO = 5 OR AO = 18 OR AO
       = 27 THEN  RETURN
3070  IF AO <  > 26 THEN 3100
3080  CC = CC + 1:
      IF CC > NC THEN CC = 1
3090  GOTO 3032
3100  IF AO <  > 23 THEN 3140
3110  CC = CC - 1:
      IF CC < 1 THEN CC = NC
3120  IF TP(CC) = 4 THEN 3110
3130  GOTO 3032
3140  IF TP(CC) <  > 1 THEN B$(CC)
       = C$:
      GOTO 3080
3150  PD = 0:
      FOR I = 1 TO LC(CC):
         C =  ASC ( MID$ (C$,I,1)):
         IF (C < 48 OR C > 57) AND C
          <  > 32 AND C <  > 46 AND C
          <  > 45 THEN B$(CC) = C$:
         PRINT BP$;:
         GOTO 3032
3180     IF C = 46 THEN PD = PD + 1
3190  NEXT I:
      IF PD > 1 THEN B$(CC) = C$:
      PRINT BP$;:
      GOTO 3032
3200  C$ =  MID$ (C$,1,IC):
      Z$ = C$:
      GOSUB 2500:
      C$ = Z$:
      KO = LC(CC) -  LEN (C$):
      A$ =  MID$ (BA$,1,KO):
      A$ = A$ + C$:
      B$(CC) = A$:
      A$ =  MID$ (BE$,1,KO):
      A$ = A$ + C$:
      HTAB XC:
      VTAB YC:
      PRINT A$:
      GOTO 3080
3500  REM
3530  IC = 0:
      C$ = B$(CC):
      IF LO = 0 THEN  RETURN
3535  IF CC = 0 THEN  HTAB XC:
      VTAB YC:
      PRINT  MID$ (BU$,1,LO)
3540  X = XC:
      Y = YC
3545  IF  MID$ (C$,IC + 1,1) = "."
       AND TP(CC) = 1 THEN IC = IC
       + 1:
      X = X + 1
3550  GOSUB 1000:
      GOSUB 1200:
      GOSUB 1100
3560  IF AO = 5 OR AO = 23 OR AO
       = 26 OR AO = 18 OR AO = 27
       THEN  RETURN
3565  IF AO = 9 THEN  GOSUB 3700:
      GOTO 3560
3567  IF AO = 4 THEN  GOSUB 3800:
      GOTO 3550
3570  IF AO = 25 THEN  GOSUB 13000:
      GOTO 3550
3580  IF AO <  > 13 THEN 3620
3582  IF IC = 0 THEN 3590
3585  IF TP(CC) = 1 AND  MID$ (C$,IC,1)
       = "." THEN IC = IC - 1
3590  C$ =  MID$ (C$,1,IC):
      A$ =  MID$ (BA$,1,LO - IC):
      C$ = C$ + A$:
      A$ =  MID$ (BU$,1,LO - IC):
      IF TP(CC) = 1 THEN A$ =  MID$ (BE
      $,1,LO - IC)
3605  HTAB XC + IC:
      PRINT A$:
      RETURN
3620  IF AO <  > 1 THEN 3650
3630  IF IC = 0 THEN  PRINT BP$;:
      GOTO 3550
3640  IC = IC - 1:
      X = X - 1:
      GOTO 3550
3650  IF IC = LO THEN  PRINT BP$;:
      GOTO 3550
3655  IF AO = 0 THEN 3550
3660  IF AO = 19 THEN IC = IC + 1:
      X = X + 1:
      GOTO 3545
3670  CH$ =  CHR$ (AO):
      PRINT CH$;:
      C$ =  MID$ (C$,1,IC) + CH$
       +  MID$ (C$,IC + 2,39):
      IC = IC + 1:
      X = X + 1:
      GOTO 3545
3700  REM
3722  IF IC = LO OR TP(CC) = 1 THEN AO
       = 0:
      RETURN
3725  Z$ = C$:
      GOSUB 2500:
      C$ = Z$
3730  FC = 0:
      GOSUB 1000:
      GOSUB 1200:
      GOSUB 1100:
      IF AO = 5 OR AO = 4 OR AO = 23
       OR AO = 26 OR AO = 18 OR AO
       = 27 OR AO = 25 OR AO = 1
       OR AO = 19 OR AO = 13 THEN FC
       = 1:
      RETURN
3750  CH$ =  CHR$ (AO):
      C$ =  MID$ (C$,1,IC) + CH$
       +  MID$ (C$,IC + 1,LO - IC
       - 1):
      L =  LEN (C$):
      A$ =  MID$ (BU$,1,LO - L):
      A$ = C$ + A$:
      HTAB XC:
      PRINT A$:
      X = X + 1:
      IC = IC + 1:
      IF IC < LO THEN 3730
3790  FC = 1:
      RETURN
3800  REM
3830  IF IC = LO OR TP(CC) = 1 THEN
       RETURN
3840  Z$ = C$:
      GOSUB 2500:
      C$ = Z$:
      C$ =  MID$ (C$,1,IC) +  MID$ (C$,
      IC + 2,LO - IC - 1):
      L =  LEN (C$):
      A$ =  MID$ (BU$,1,LO - L):
      A$ = C$ + A$:
      HTAB XC:
      PRINT A$:
      RETURN
4500  REM --- INTRODUCCION DATOS ---
4530  GOSUB 7000
4540  GOSUB 500:
      HTAB 4:
      VTAB 14:
      PRINT "1) EL FICHERO EXISTE":
      HTAB 4:
      VTAB 16:
      PRINT "2) EL FICHERO DEBE CREARSE
      ":
      HTAB 11:
      VTAB 19:
      PRINT "CUAL ?";
4560  GET A$:
      IF A$ <  > "1" AND A$ <  > "2"
       THEN 4560
4570  PRINT A$;:
      FOR I = 1 TO 400:
      NEXT
4580  IF A$ = "1" THEN  GOSUB 5200:
      GOTO 4600
4590  MX = 0:
      GOTO 4650
4600  MX =  VAL (DA$(0))
4650  REM - PRINCIPIO BUCLE INTRODUCION
4660  FOR KJ = MX + 1 TO MR:
         H = KJ
4670     GOSUB 2000:
         HTAB 3:
         VTAB 24:
         INVERSE :
         PRINT "REGISTRO N.";H;:
         NORMAL :
         GOSUB 3000
4680     IF AO = 27 THEN 4670
4685     IF AO = 18 THEN  RETURN
4690     DA$(KJ) = "*"
4700     FOR I = 1 TO M3:
            IF TP(I) = 0 THEN 4720
4710        DA$(KJ) = DA$(KJ) + B$(I)
4720     NEXT I
4730     DA$(0) =  STR$ (KJ)
4740     GOSUB 500:
         HTAB 8:
         VTAB 12:
         PRINT "OTRA INTRODUCCION ? (S/
         N)";
4750     GET A$:
         IF A$ = "N" THEN 4780
4760     IF A$ <  > "S" THEN 4750
4770  NEXT KJ:
      KJ = KJ - 1
4780  GOSUB 5300:
      RETURN
5000  REM ----------------------------
5010  PRINT D$:
      PRINT D$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2:
      PRINT D$:
      RETURN
5020  REM ----------------------------
5030  PRINT D$:
      PRINT D$;"READ";NF$;",R";NR:
      INPUT AA$:
      PRINT D$:
      RETURN
5040  REM ----------------------------
5050  PRINT D$:
      PRINT D$;"WRITE";NF$;",R";NR:
```

```
        PRINT  CHR$ (34) + AA$:
        PRINT D$:
        RETURN
5060    REM ----------------------------
5070    PRINT D$:
        PRINT D$;"CLOSE";NF$:
        PRINT D$:
        RETURN
5200    REM  --- LECTURA ---
5210    IF O6 = 2 THEN  RETURN
5220    O6 = 2
5230    GOSUB 500:
        HTAB 10:
        VTAB 13:
        PRINT "INTRODUCIR EL DISCO DATOS"
        :
        HTAB 10:
        VTAB 15:
        PRINT "DESPUES PULSAR UNA TECLA":
        GET A$:
        GOSUB 500:
        HTAB 5:
        VTAB 13:
        PRINT "NOMBRE FICHERO";:
        INPUT NF$
5240    GOSUB 500:
        GOSUB 600:
        LR = 128:
        GOSUB 5000
5250    NR = 1:
        GOSUB 5020:
        DA$(0) = AA$
5260    FOR I = 1 TO  VAL (DA$(0)):
           NR = I + 1
5270       GOSUB 5020:
           DA$(I) = AA$:
        NEXT
5280    GOSUB 5060:
        RETURN
5300    REM  --- GRABACION DATOS ---
5330    GOSUB 500:
        HTAB 10:
        VTAB 13:
        PRINT "INTRODUCIR EL DISCO DATOS"
        :
        HTAB 10:
        VTAB 15:
        PRINT "DESPUES PULSAR UNA TECLA":
        GET A$:
        GOSUB 500:
        HTAB 5:
        VTAB 13:
        PRINT "NOMBRE DEL FICERO";:
        INPUT NF$
5340    GOSUB 500:
        GOSUB 600:
        LR = 128
5350    GOSUB 5000:
        FOR I = 0 TO  VAL (DA$(0))
5360       AA$ = DA$(I):
           NR = I + 1:
           GOSUB 5040:
        NEXT I
5370    GOSUB 5060
5375    O6 = 2
5380    RETURN
5500    REM  --- ORDENADO ---
5540    GOSUB 7000:
        GOSUB 5200
5550    FOR I = 1 TO M3
5560       HT(I) = TP(I):
           HL(I) = LC(I):
           HN(I) = ND(I)
5570       IF TP(I) <  > 0 THEN TP(I)
           = 3:
           LC(I) = 1:
           ND(I) = 0
5580    NEXT I
5590    GOSUB 2000:
        HTAB 2:
        VTAB 23:
        PRINT "'*' EN EL CAMPO CLAVE DEL
        ORDENADO";BP$;:
        GOSUB 3000
5600    IF AO = 27 THEN 5590
5605    FOR I = 1 TO M3:
           TP(I) = HT(I):
           ND(I) = HN(I):
           LC(I) = HL(I):
        NEXT I
5608    IF AO = 18 THEN  RETURN
5610    FOR I = 1 TO M3:
           IF B$(I) = "*" THEN K4 = I:
           GOTO 5630
5620    NEXT I:
        GOSUB 500:
        HTAB 3:
        VTAB 11:
        PRINT "EL CAMPO CLAVE NO SE HA IN
        DICADO":
        HTAB 13:
        VTAB 14:
        PRINT "VUELVO AL MENU";BP$:
        FOR I = 1 TO 1000:
        NEXT :
        RETURN
5630    IF K4 = 1 THEN P1 = 2:
        P2 = LC(1):
        GOTO 5670
5640    P1 = 1:
        P2 = 0:
        FOR I = 1 TO K4 - 1
5650       P1 = P1 + LC(I):
        NEXT
5660    P1 = P1 + 1:
        P2 = LC(K4)
5670    GOSUB 500:
        GOSUB 600:
        GOSUB 6000:
        W6 = 2:
        P8 = P1:
        P9 = P2
5680    GOSUB 500:
        HTAB 10:
        VTAB 12:
        PRINT "ORDENADO EFECTUADO";BP$:
        FOR I = 1 TO 1000:
        NEXT
5690    RETURN
6000    REM  SORT
6010    N =  VAL (DA$(0))
6020    FOR I = 1 TO N:
           IS(I) = I:
        NEXT
6040    I1 = 1:
        J1 = N
6050    I = I1:
        J = J1:
        S =  - 1
6060    GOSUB 6220
6065    IF TP(K4) < 3 THEN  IF  VAL (A1$)
        <  =  VAL (B1$) THEN 6100
6067    IF TP(K4) < 3 THEN 6080
6070    IF A1$ <  = B1$ THEN 6100
6080    V = IS(I):
        IS(I) = IS(J)
6090    IS(J) = V:
        S =  SGN ( - S)
6100    IF S = 1 THEN I = I + 1:
        GOTO 6120
6110    J = J - 1
6120    IF I < J THEN 6060
6130    IF I + 1 > J1 THEN 6160
6140    P = P + 1
6150    S9(P,1) = I + 1:
        S9(P,2) = J1
6160    J1 = I - 1
6170    IF I1 < J1 THEN 6050
6180    IF P = 0 THEN 6210
6190    I1 = S9(P,1):
        J1 = S9(P,2):
        P = P - 1
6200    GOTO 6050
6210    RETURN :
        REM --- FIN ORDENADO ---

6220    A1$ =  MID$ (DA$(IS(I)),P1,P2)
6230    B1$ =  MID$ (DA$(IS(J)),P1,P2)
6240    RETURN
7000    REM -- LECTURA DATOS MASCARA --
7025    IF O8 = 2 THEN  RETURN
7027    O8 = 2
7030    GOSUB 500:
        HTAB 5:
        VTAB 13:
        PRINT "INTRODUCIR EL DISCO DEFINI
        CIONES":
        HTAB 10:
        VTAB 16:
        PRINT "DESPUES PULSAR UNA TECLA":
        GET A$:
        GOSUB 600
7040    NF$ = "DEFC":
        LR = 26:
        GOSUB 5000
7050    FOR I = 1 TO M3:
           NR = I:
           GOSUB 5020
7060       XD(I) =  VAL ( LEFT$ (AA$,2)):
           YD(I) =  VAL ( MID$ (AA$,3,2))
           :
           D$(I) =  MID$ (AA$,5,10)
7080       XC(I) =  VAL ( MID$ (AA$,15,2)
           ):
           YC(I) =  VAL ( MID$ (AA$,17,2)
           )
7090       LC(I) =  VAL ( MID$ (AA$,19,2)
           ):
           TP(I) =  VAL ( MID$ (AA$,21,2)
           ):
           ND(I) =  VAL ( MID$ (AA$,23,2)
           )
7100    NEXT I:
        GOSUB 5060
7110    NF$ = "FORC":
        LR = 34:
        GOSUB 5000
7120    FOR I = 1 TO M3:
           NR = I:
           GOSUB 5020
7140       F$(I) =  MID$ (AA$,3,30)
7150    NEXT I:
        GOSUB 5060
7160    NF$ = "PINC":
        LR = 82:
        GOSUB 5000
7170    NR = 1:
        GOSUB 5020
7180    XP =  VAL ( LEFT$ (AA$,2)):
        II$ =  MID$ (AA$,3,80)
7190    GOSUB 5060:
        NF$ = "PDFC":
        LR = 82
7200    GOSUB 5000:
        NR = 1:
        GOSUB 5020
7210    FI =  VAL ( LEFT$ (AA$,2))
7220    C1 = 3:
        FOR I = 1 TO M3:
           FT(I) =  VAL ( MID$ (AA$,C1,2)
           ):
           NS(I) =  VAL ( MID$ (AA$,C1
           + 2,2)):
           C1 = C1 + 4
7230    NEXT
7240    L =  LEN (II$):
        IF L = 0 THEN  RETURN
```

```
7250 FOR I =  LEN (II$) TO 1 STEP
      - 1
7260    IF  MID$ (II$,I,1) <  > " "
      THEN 7265
7262    L = L - 1:
     NEXT I
7265 II$ =  LEFT$ (II$,L)
7270 RETURN
8500 REM --- PREPARA FORMULA ---
8520 A$ = F$(CC)
8530 L =  LEN (A$):
     FOR I = 1 TO L
8540    IF  MID$ (A$,I,1) =  CHR$ (39)
      THEN 8560
8550 NEXT I:
     GOTO 8620
8560 A1$ =  MID$ (A$,I + 1,1):
     IF  MID$ (A$,I + 2,1) <  >
      CHR$ (39) THEN A1$ = A1$ +
      MID$ (A$,I + 2,1):
     P = 4:
     GOTO 8570
8565 P = 3
8570 A =  VAL (A1$):
     A2$ = B$(A)
8590 IF  VAL (B$(A)) = 0 THEN A2$
      = "0"
8600 A$ =  MID$ (A$,1,I - 1) + A2$
      +  MID$ (A$,I + P,L)
8610 GOTO 8530
8620 GOTO 9000
8625 IF ER <  > 0 THEN B$(CC) = "$":
     L = 1:
     GOSUB 8700:
     PRINT BP$;:
     CC = 0:
     GOTO 8695
8630 L =  LEN (A$):
     PP = 1
8640 FOR K1 = 1 TO L:
       IF  MID$ (A$,K1,1) = "."
        THEN 8657
8650 NEXT K1:
     A1$ = A$:
     A2$ = ".":
     GOTO 8660
8657 A1$ =  LEFT$ (A$,K1 - 1):
     IF ND(CC) <  > 0 THEN A2$ =
      MID$ (A$,K1,ND(CC) + 1)
8660 IF ND(CC) = 0 THEN A2$ = "":
     PP = 0:
     GOTO 8670
8665 IF ( LEN (A2$) - 1) < ND(CC)
      THEN A2$ = A2$ +  MID$ (ZE$,1,ND
      (CC) -  LEN (A2$) + 1)
8670 N = LC(CC) - ND(CC) - PP:
     IF  LEN (A1$) > N THEN B$(CC)
      = "$":
     L = 1:
     GOSUB 8700:
     PRINT BP$;:
     CC = 0:
     GOTO 8695
8680 B$(CC) = A1$ + A2$:
     L =  LEN (B$(CC))
8690 GOSUB 8700
8695 GOTO 3037
8700 A$ =  MID$ (BE$,1,LC(CC) - L)
      + B$(CC)
8705 HTAB XC(CC):
     VTAB YC(CC)
8710 PRINT A$:
     B$(CC) =  MID$ (BA$,1,LC(CC)
      - L) + B$(CC)
8715 RETURN
9000 REM  --- CALCULO ---
9040 Z$ = A$:
     GOSUB 2900:
     A$ = "+" + Z$
```

```
9070 GOTO 9350
9080 REM  *** CAL 2
9090 ER = 0:
     L =  LEN (CA$):
     IF L < 3 THEN 9335
9100 FOR I = 2 TO L - 1:
       CH$ =  MID$ (CA$,I,1):
       FOR JJ = 1 TO 5:
         IF OP$(JJ) = CH$ THEN 9120
9110 NEXT JJ,I:
     ER = 1:
     GOTO 9335
9120 CH$ =  MID$ (CA$,I + 1,1):
     IF CH$ = "*" OR CH$ = "^" OR CH$
      = "/" THEN ER = 2:
     GOTO 9335
9130 Z1 =  VAL ( MID$ (CA$,1,I - 1)):
     Z2 =  VAL ( MID$ (CA$,I + 1,40)):
     ON JJ GOSUB 9150,9160,9180,9190,9
     200:
     CA$ =  STR$ (Z1):
     IF Z1 >  = 0 THEN CA$ = "+"
      + CA$
9140 GOTO 9335
9150 Z1 = Z1 ^ Z2:
     RETURN
9160 IF Z2 = 0 THEN ER = 3:
     RETURN
9170 Z1 = Z1 / Z2:
     RETURN
9180 Z1 = Z1 * Z2:
     RETURN
9190 Z1 = Z1 - Z2:
     RETURN
9200 Z1 = Z1 + Z2:
     RETURN
9210 REM  *** FORM 2
9220 ER = 0
9230 FOR OP = 1 TO 5:
       LO =  LEN (FO$):
       KK = 2:
       IF LO = 1 THEN KK = 1
9240   PO = 0:
       P1 = 0:
       P2 = 0:
       FOR IO = KK TO LO - 1:
         CH$ =  MID$ (FO$,IO,1):
         CH =  ASC (CH$):
         IF CH$ = OP$(OP) AND PO
          = 0 THEN PO = IO:
         GOTO 9280
9250     O = 0:
         IF (CH = 94 OR CH < 48)
          AND CH <  > 46 THEN O
          = 1
9260     IF O = 1 AND PO = 0 THEN P1
          = IO
9270     IF O = 1 AND PO > 0 AND IO
          - PO > 1 THEN P2 = IO:
         GOTO 9300
9280   NEXT IO:
       IF PO > 0 THEN 9300
9290 NEXT OP:
     RETURN
9300 IF P1 = 0 THEN P1 = 1:
     GOTO 9320
9310 CH$ =  MID$ (FO$,P1,1):
     IF CH$ <  > "+" AND CH$ <
      > "-" THEN P1 = P1 + 1
9320 P2 = P2 - 1:
     IF P2 < 0 THEN P2 = LO
9330 CA$ =  MID$ (FO$,P1,P2 - P1
      + 1):
     GOTO 9090:
     IF ER <  > 0 THEN  RETURN
9335 IF ER <  > 0 THEN  RETURN
9340 GOSUB 9640:
     FO$ =  MID$ (FO$,1,P1 - 1)
      + CA$ +  MID$ (FO$,P2 + 1,40):
```

```
     GOTO 9220
9350 REM  *** PAR 1
9360 ER = 0:
     LA =  LEN (A$):
     IF LA < 3 THEN 9410
9370 PI = 0:
     PF = 0:
     FOR IA = 1 TO LA:
       CH$ =  MID$ (A$,IA,1):
       IF CH$ = "(" THEN PI = IA
9380   IF CH$ = ")" THEN 9450
9390 NEXT IA:
     IF PI > 0 THEN ER = 4:
     GOTO 8625
9400 FO$ = A$:
     GOSUB 9210:
     A$ = FO$
9410 IF  VAL (A$) = 0 THEN A$ = "0"
9420 IF  MID$ (A$,2,1) = "." THEN A$
      =  LEFT$ (A$,1) + "0" +  MID$ (A
     $,2,80)
9430 IF  LEFT$ (A$,1) = "+" THEN A$
      =  MID$ (A$,2,80)
9440 GOTO 8625
9450 IF PI = 0 THEN ER = 4:
     GOTO 8625
9460 PF = IA:
     FO$ =  MID$ (A$,PI + 1,PF - PI
      - 1):
     GOSUB 9210:
     IF ER <  > 0 THEN 8625
9470 IF PI > 3 THEN  GOSUB 9520
9480 IF PI <  = 1 THEN 9500
9490 IF A < 0 AND  MID$ (A$,PI - 1,1)
      = "-" THEN PI = PI - 1:
     FO$ = "+" +  MID$ (FO$,2,80)
9500 IF ER <  > 0 THEN 8625
9510 A$ =  MID$ (A$,1,PI - 1) + FO$
      +  MID$ (A$,PF + 1,80):
     GOTO 9360
9520 REM  *** FUN 1
9530 ER = 0:
     CH$ =  MID$ (A$,PI - 1,1):
     IF CH$ < "A" OR CH$ > "Z" THEN
      RETURN
9540 FU$ =  MID$ (A$,PI - 3,3):
     FOR I = 1 TO 6:
       IF FU$ = FZ$(I) THEN 9560
9550 NEXT I:
     ER = 5:
     RETURN
9560 ZA =  VAL (FO$):
     ON I GOSUB 9570,9580,9590,9600,96
     10,9630:
     CA$ =  STR$ (ZA)
9565 IF ZA >  = 0 THEN CA$ = "+"
      + CA$
9567 GOSUB 9640:
     FO$ = CA$:
     PI = PI - 3:
     RETURN
9570 ZA =  ABS (ZA):
     RETURN
9580 ZA =  INT (ZA):
     RETURN
9590 ZA =  COS (ZA):
     RETURN
9600 ZA =  SIN (ZA):
     RETURN
9610 IF ZA < 0 THEN ER = 6:
     RETURN
9620 ZA =  SQR (ZA):
     RETURN
9630 ZA =  TAN (ZA):
     RETURN
9640 REM  *** EXP 2
9650 L =  LEN (CA$)
9660 FOR IC = 1 TO L:
       IF  MID$ (CA$,IC,1) = "E"
```

```
      THEN 9680
9670 NEXT :
     RETURN
9680 N$ =  MID$ (CA$,2,IC - 2):
     L =  LEN (N$)
9690 FOR I = 1 TO L:
       IF  MID$ (N$,I,1) = "."
        THEN P = I:
       GOTO 9710
9700 NEXT :
     P = L
9710 E =  VAL ( MID$ (CA$,IC + 1,3)):
     IF E < 0 THEN 9790
9720 DD = L - P:
     IF DD = 0 THEN 9770
9730 N$ =  MID$ (N$,1,P - 1) +
      MID$ (N$,P + 1,80)
9740 IF DD <  = E THEN 9770
9750 N$ =  MID$ (N$,1,P + E - 1)
      + "." +  MID$ (N$,P + E,80)
9760 DD = E
9770 N$ = N$ +  MID$ (ZE$,1,E - DD)
9780 GOTO 9830
9790 IF P = L THEN P = P + 2 - L:
     GOTO 9810
9800 N$ =  MID$ (N$,1,P - 1) +
      MID$ (N$,P + 1,80):
     L = 1
9810 N$ =  MID$ (ZE$,1, - E - L
      + 1) + N$
9820 N$ =  MID$ (N$,1,P - 1) + "."
      +  MID$ (N$,P,80)
9830 CA$ =  LEFT$ (CA$,1) + N$
9840 RETURN
10000 REM --- DATOS ---
10010 DATA 7
10020 DATA "DEFINICION MASCARA","INTROD
      UCCION DATOS","ORDENADO","BUSQUED
      A CON VARIACION","DEFINICION FORM
      ATOS IMPRESION","IMPRESION","FIN
      DE TAREA"
10030 DATA "^","/","*","-","+"
10040 DATA "ABS","INT","COS","SIN","SQR
      ","TAN"
11000 REM -------------------
11010 TI$ = "PULSAR UNA TECLA":
      YT = 22:
      GOSUB 2700:
      GOSUB 1200:
      GOSUB 2800:
      RETURN
12000 REM --- DEFINICION MASCARA   ---
12005 REM --- Y FORMATOS IMPRESION ---
12010 GOSUB 500:
      HTAB 7:
      VTAB 13:
      PRINT "INTRODUCIR EL DISCO PROGRA
      MAS"
12020 HTAB 10:
      VTAB 15:
      PRINT "DESPUES PULSAR UNA TECLA "
      ;:
      GET A$
12040 PRINT D$:
      PRINT D$;"RUN DEFINICIONES.FUENTE
      ,D1"
13000 REM ---------------------
13030 YT = 23:
      GOSUB 2800:
      INVERSE
13040 FOR I = 1 TO 166:
        FOR T = 1 TO 100:
        NEXT T:
        CH$ =  MID$ (HE$,I,1)
13060   A$ =  RIGHT$ (A$,38) + CH$:
        HTAB 1:
        VTAB 23:
        PRINT A$;:
        AO =  PEEK ( - 16384):
        IF AO > 128 THEN 13130
13100 NEXT I:
      GOTO 13040
13130 NORMAL :
      HTAB 1:
      VTAB 23:
      PRINT  SPC( 39);:
      POKE  - 16368,0:
      RETURN
14000 REM - BUSQUEDA CON VARIACION -
14030 GOSUB 500:
      HTAB 10:
      VTAB 5:
      PRINT  LEFT$ (BU$, LEN (VO$(4)))
14040 HTAB 10:
      VTAB 6:
      PRINT VO$(4):
      HTAB 10:
      VTAB 7
14050 PRINT  LEFT$ (BU$, LEN (VO$(4)))
14070 HTAB 6:
      VTAB 13:
      PRINT "INTRODUCIR EL NUMERO DEL R
      EGISTRO        (0= BUSQUEDA EN CAM
      PO CLAVE)              (-1 = VUELVE
      AL MENU)"
14080 HTAB 17:
      VTAB 21:
      INPUT A$
14090 R =  VAL (A$):
      IF R >  VAL (DA$(0)) THEN 14030
14100 IF R < 0 THEN  RETURN
14110 IF R <  > 0 THEN W6 = 0:
      MI = R:
      MA = R:
      GOTO 14250
14120 IF W6 <  > 2 THEN  GOSUB 500:
      HTAB 3:
      VTAB 12:
      PRINT "PARA PETICION DE BUSQUEDA
                  ANTES HAY QUE HACER
      EL ORDENADO.";BP$:
      FOR I = 1 TO 1800:
      NEXT :
      GOTO 220
14130 FOR Y = 11 TO 22:
        HTAB 1:
        VTAB Y:
        PRINT  SPC( 40);:
      NEXT
14140 HTAB 3:
      VTAB 12:
      PRINT "INTRODUCIR EL VALOR DEL CA
      MPO CLAVE"
14150 HTAB 2:
      VTAB 16:
      PRINT "NOTA. EL CAMPO CLAVE DEBE
      SER EL MISMO   QUE EL DE ORDENADO
      ."
14160 HTAB 7:
      VTAB 20:
      INPUT RA$
14162 IF  LEN (RA$) > LC(K4) THEN 14130

14165 GOSUB 500:
      HTAB 9:
      VTAB 7:
      PRINT "ESPERAR,ESTOY BUSCANDO.":
      X = (40 -  LEN (RA$)) / 2:
      HTAB X:
      VTAB 8:
      PRINT RA$
14170 IF TP(K4) = 3 THEN RA$ = RA$
       +  MID$ (BA$,1,LC(K4) -  LEN (RA
      $)):
      GOTO 14190
14180 RA$ =  MID$ (BA$,1,LC(K4) -
       LEN (RA$)) + RA$
14190 GOSUB 14500:
      REM --- BUSQUEDA BIN.---

14195 FOR I = 1 TO 1500:
      NEXT
14200 HTAB 2:
      VTAB 7:
      PRINT  SPC( 38);BP$
14210 IF TR = 0 THEN  HTAB 12:
      VTAB 14:
      PRINT "N O   H A L L A D O";BP$;:
      FOR I = 1 TO 2000:
      NEXT :
      GOTO 14000
14220 A = MA - MI + 1:
      HTAB 14:
      VTAB 14:
      PRINT "H A L L A D O":
      HTAB 13:
      VTAB 20
14230 IF A = 1 THEN  PRINT "(UN SOLO RE
      GISTRO)"
14235 IF A <  > 1 THEN  PRINT "( ";A;"
      REGISTROS)"
14240 PRINT BP$;:
      FOR I = 1 TO 2500:
      NEXT
14250 REM --- PRES REG.---
14260 FOR JK = MI TO MA:
        KJ = IS(JK)
14265   IF R <  > 0 THEN KJ = JK
14270   A$ =  MID$ (DA$(KJ),2, LEN (DA
        $(KJ)) - 1)
14280   C1 = 1:
        FOR I = 1 TO M3:
          IF TP(I) = 0 THEN 14300
14290     B$(I) =  MID$ (A$,C1,LC(I))
          :
          C1 = C1 + LC(I):
          HT(I) = TP(I):
          TP(I) = 4
14300   NEXT I
14310   H = JK:
        IF R = 0 THEN H = KJ
14320   GOSUB 2000:
        HTAB 3:
        VTAB 24:
        PRINT "REGISTRO N.";H;
14330   FOR I = 1 TO M3:
          TP(I) = HT(I):
        NEXT :
        IF R = 0 THEN TP(K4) = 4
14340   GOSUB 3000:
        TP(K4) = HT(K4)
14350   IF AO = 27 THEN 14270
14355   IF AO = 18 THEN  RETURN
14360   DA$(KJ) = "*"
14370   FOR I = 1 TO M3:
          IF TP(I) = 0 THEN 14390
14380     DA$(KJ) = DA$(KJ) + B$(I)
14390   NEXT I
14400   HTAB 2:
        VTAB 23:
        PRINT "PULSAR UNA TECLA
            ";
14410   GET A$
14420 NEXT JK:
      GOSUB 500:
      HTAB 8:
      VTAB 12:
      PRINT "OTRA VARIACION ? (S/N)";
14430 GET A$:
      IF A$ = "S" THEN 14000
14440 IF A$ <  > "N" THEN 14430
14450 GOSUB 5300:
      RETURN
14500 REM --- BUSQUEDA BINARIA ---
14515 P1 = P8:
      P2 = P9
14520 RI = 0:
```

```
      RF =  VAL (DA$(0)):
      RM =  INT (RF / 2)
14530 RE = IS(RM):
      RB$ =  MID$ (DA$(RE),P1,P2)
14540 IF RA$ = RB$ THEN 14640
14550 IF RA$ > RB$ THEN 14600
14560 RF = RM:
      RM =  INT ((RF - RI) / 2)
14570 IF RM < 1 THEN RM = 1
14580 RM = RM + RI:
      IF RM >  = RF THEN TR = 0:
      RETURN
14590 GOTO 14530
14600 RI = RM:
      RM =  INT ((RF - RI) / 2)
14610 IF RM < 1 THEN RM = 1
14620 RM = RM + RI:
      IF RM > RF THEN TR = 0:
      RETURN
14630 GOTO 14530
14640 SG =  - 1:
      RF = RM:
      RI = RM:
      MI = RM:
      MA = RM:
      FM = 0
14650 RM = RM + SG:
      RE = IS(RM):
      RB$ =  MID$ (DA$(RE),P1,P2)
14660 IF RA$ = RB$ THEN 14690
14670 IF FM = 1 THEN TR = 1:
      RETURN
14680 FM = 1:
      SG = 1:
      RM = RF:
      GOTO 14650
14690 IF FM = 1 THEN MA = MA + 1:
      GOTO 14700
14695 MI = MI - 1
14700 GOTO 14650
16000 REM --- IMPRIME ---
16030 IF W6 <  > 2 THEN  GOSUB 500:
      HTAB 2:
      VTAB 10:
      PRINT "ANTES DE LA IMPRESION HAY
      QUE HACER "; CHR$ (10);"    EL OR
      DENADO.   VUELVO AL MENU";BP$;:
      FOR I = 1 TO 2500:
      NEXT :
      RETURN
16045 GOSUB 500:
      GOSUB 600
16047 PRINT D$:
      PRINT D$;"PR#1":
      PRINT D$
16048 PRINT  CHR$ (9) + "80N";
16049 GOSUB 17000:
      CT = 0
16050 FOR J = 1 TO M3:
        Z8(J) = 0:
      NEXT
16055 FOR I = 1 TO MR
16060   IF IS(I) = 0 THEN 16200
16065   IF CT > 35 THEN CT = 0:
        PRINT  CHR$ (12):
        GOSUB 17000
16070   A$ =  MID$ (DA$(IS(I)),2,200)
16080   C1 = 1:
        FOR J = 1 TO M3:
          IF TP(J) = 0 THEN 16100
16090     B$(J) =  MID$ (A$,C1,LC(J))
          :
          C1 = C1 + LC(J)
16100   NEXT J
16105   IF FI = 2 THEN  PRINT "-------
        ------":
        PRINT "REGISTRO N.";IS(I):
        PRINT "-------------"
16110   A$ = "*":
        FOR J = 1 TO M3:
          IF NS(J) = 0 THEN 16150
16120     IF FT(J) = 1 THEN Z8(J)
           = Z8(J) +  VAL (B$(NS(J)))

16130     IF FI = 1 THEN A$ = A$
           + B$(NS(J)) + "*":
          GOTO 16150
16140     PRINT B$(NS(J)):
          CT = CT + 1
16150   NEXT J
16160   IF FI = 1 THEN  PRINT A$:
        CT = CT + 1:
        GOTO 16170
16165   PRINT BU$;BU$:
        CT = CT + 4
16170 NEXT I
16200 REM --- TOTAL ---
16210 PRINT  TAB( 1);BU$;BU$:
      KK = 0
16220 IF FI = 2 THEN 16300
16230 A = 1:
      FOR I = 1 TO M3:
        IF TP(I) = 0 THEN 16260
16240   A = A + LC(NS(I)) + 1:
        IF FT(I) = 0 THEN 16260
16245   AA = A -  LEN ( STR$ (Z8(I))):
        IF AA < 1 THEN AA = 1
16247   IF AA - KK < 0 THEN KK = AA
16250   PRINT  TAB( AA - KK);Z8(I);:
        KK = AA +  LEN ( STR$ (Z8(I)))
         - 1
16260 NEXT I:
      PRINT
16280 GOTO 16350
16300 PRINT  TAB( 10);"T O T A L E S":
      PRINT  TAB( 10);"*************"
16310 FOR I = 1 TO M3:
        IF TP(I) = 0 THEN 16340
16320   IF FT(I) = 0 THEN  PRINT :
        GOTO 16340
16330   PRINT  TAB( 11);Z8(I)
16340 NEXT I
16350 PRINT  CHR$ (12):
      PRINT D$:
      PRINT D$;"PR#0":
      PRINT D$:
      RETURN
17000 REM --- CABECERA ---
17010 PRINT BU$;BU$:
      PRINT
17020 PRINT  TAB( XP + 1);II$
17030 PRINT  TAB( 1);BU$;BU$
17040 RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - APPLE II

Definiciones

LC(•)	= longitud de los campos de introducción
ND(•)	= número de decimales en el campo de introducción
XC(•) YC(•)	= coordenadas en el vídeo del campo de introducción
XD(•) YD(•)	= coordenadas en el vídeo de la descripción
D$(•)	= descripciones en la máscara vídeo
B$ TP(•)	= matriz que contiene los tipos de campos de introducción
FO$(•)	= matriz de las fórmulas de cálculo
FT$(•) NS$(•)	= utilizadas en las definiciones parámetros de impresión
H1$	= cabecera
DR$	= unidad seleccionada
D1$	= cadena de comunicación unidad
A	= código de la tecla pulsada en definición máscara
LR	= longitud registro
XP	= columna principio impresión
II$	= contiene la cabecera de la impresión
XC(•) YC(•)	= coordenadas en el vídeo del campo de introducción
XD(•) YD(•)	= coordenadas en el vídeo de la descripción
D$(•)	= descripciones en las máscaras vídeo
BL$	= blanck
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de introducción alfanuméricos
NS$	= carácter "#"; identifica los campos de introducción numéricos
NUC	= número de campos de introducción en la máscara vídeo activada
MR	= número máximo de registros
HT(•) HL(•) HN(•)	= matriz auxiliar para máscara vídeo ordenado
IS(•)	= indicadores de los registros después del ordenado
NS(•) FT(•)	= parámetros de impresión
Z8(•)	= totales en fase de impresión
F$(•)	= matriz de las fórmulas
DA$(•)	= matriz de los datos insertados
S9(•, •)	= para el ordenado
KJ	= índice del bucle de introducción de datos

Main

B$(•)	= matriz de los buffers (1 por campo de introducción)
LC(•)	= longitud de los campos de introducción
ND(•)	= número de decimales en el campo de introducción
MX	= último registro introducido
K4	= número del campo clave para el ordenado
ER	= índice de error
OP$(•)	= operadores reconocidos
FZ$(•)	= funciones reconocidas

Base de Datos/Versión Philips VG 8010

Definiciones

```
40000 REM ------------------------------
40010 REM DEFINIC. MASCARA E IMPRESION
40020 REM ------------------------------
40030 SCREEN0:KEYOFF:KEY1,CHR$(17):KEY2,CHR$(15):KEY3,CHR$(16):KEY4,"":KEY5,CHR$(20):WIDTH40:CLE
AR10000:DEFINTA-Z
40040 COLOR15,5
40050 M3=20:REM M3=MAX N. CAMPOS
40060 DIM D$(M3),TP(M3),LC(M3),ND(M3),XD(M3),YD(M3),XC(M3),YC(M3),FO$(M3),FT$(M3),NS$(M3)
40070 H1$="E.G.S.              Editorial Forum, S.A."
40200 REM -------
40210 REM M A I N
40220 REM -------
40230 GOSUB46000:REM MENU
40240 ONOPZGOSUB41000,45000,42500,43000
40250 GOTO40200
41000 GOSUB43500:CLS
41010 X=1:Y=2:GOSUB 41150:GOSUB 41020:GOTO 41070
41020 A$=CHR$(1)+CHR$(&H57):FORI=1TO38:L$=L$+A$:NEXT
41030 L1$=CHR$(1)+CHR$(&H58)+L$+CHR$(1)+CHR$(&H59):L2$=CHR$(1)+CHR$(&H5A)+L$+CHR$(1)+CHR$(&H5B):
L$=""
41040 LOCATE1,0:PRINTH1$;
41050 PRINTL1$;:FORI=1TO20:PRINT" |"+SPACE$(38)+" |";:NEXT:PRINT L2$;
41060 RETURN
41070 REM ------------------
41080 REM PRINCIPIO CREACION
41090 REM ------------------
41095 FOR I=1 TO M3:LC(I)=0:ND(I)=0:TP(I)=0:NEXT
41100 GOSUB41310
41110 A=ASC(I$):IF A<32 THEN 41180:REM comandi
41120 IFA>122THEN41100
41130 PRINTI$:X=X+1:IFX>38THENX=38:GOSUB41150ELSEGOSUB41150
41140 GOTO41100
41150 REM
41160 LOCATE3,23:PRINT"Columna:";X;TAB(16)"Linea:";Y-1;TAB(26)"F1 = AYUDA";
41170 RETURN
41180 REM --------
41190 REM COMANDOS
41200 REM --------
41210 IFA=28THENX=X+1:IFX>38THENX=1:A=31:GOTO41240
41220 IFA=29THENX=X-1:IFX<1THENX=38:A=30:GOTO41230
41230 IFA=30THENY=Y-1:IFY<2THENY=2
41240 IFA=31THENY=Y+1:IFY>21THENY=21
41250 IFA=13THENX=1:Y=Y+1:IFY>21THENY=21
41260 IFA=15THENGOSUB41360
41270 IFA=16THENGOSUB41400
41280 IFA=20THENGOTO41460
41290 IFA=17THENGOSUB44000
41300 GOSUB41150:GOTO41100
41310 REM -- acepta campo ---
41320 I$=INKEY$:IFI$<>""THENLOCATEX,Y,0:RETURN
41330 T=T+1:IFT<40THEN41320
41340 IFC=1THENC=0ELSEC=1
41350 LOCATEX,Y,C:T=1:GOTO41320
41360 REM --- INSERCION ---
41370 P1=0:P2=Y*40+X+1:P3=(Y+1)*40-2
41380 FORI=P3TOP2STEP-1:LOCATE38-P1,Y:PRINTCHR$(VPEEK(I-1));:P1=P1+1:NEXT:LOCATEX,Y:PRINT" ";
41390 RETURN
41400 REM -------
41410 REM BORRADO
41420 REM -------
41430 P1=0:P2=Y*40+X:P3=(Y+1)*40-2
41440 FORI=P2TOP3:LOCATEX+P1,Y:PRINTCHR$(VPEEK(I+1));:P1=P1+1:NEXT:PRINT" "
41450 RETURN
41460 REM -------------
41470 REM LECTURA VIDEO
41480 REM -------------
41490 LOCATE0,23:PRINTSPACE$(13);"E S P E R A R";SPACE$(10);:PLAY"O7L32CDEF"
41500 YY=1:PD=0:PC=0
41510 FORJJ=2TO21:J=JJ*40:YY=YY+1
41520 FORK=1TO38:XX=XX+1
41530 A$=CHR$(VPEEK(K+J))
41540 IFA$="&"THEN41660
41550 IFA$="X"ORA$="N"ORA$="C"THEN41570
41560 IFA$<>" "THEN41770ELSE41730
41570 PC=PC+1:IFPC>M3THEN41810
41580 IFA$="X"THENTP(PC)=3ELSEIFA$="N"THENTP(PC)=1ELSETP(PC)=2
41590 LC(PC)=1:YC(PC)=YY:XC(PC)=K:ND(PC)=0:W6=0
41600 K=K+1:IFK=40THEN41770
41610 IFW6=1ANDA$="X"THEN41770
41620 IFCHR$(VPEEK(J+K))=A$THENLC(PC)=LC(PC)+1:IFW6=1THENND(PC)=ND(PC)+1:GOTO41600ELSEGOTO41600
41630 V=VPEEK(J+K):IFV=46ANDW6=0THENLC(PC)=LC(PC)+1:W6=1:GOTO41600
41640 IFV=32ORV=22ORV=27ORV=25THENK=K-1:GOTO41730
41650 IFV<>38THEN41770
41660 REM --- descripciones ---
41670 PD=PD+1:IFPD>M3THEN41810
41680 D$(PD)="":XD(PD)=K+1:YD(PD)=YY
41690 K=K+1:IFK=39THEN41770:REM ERRORE
41700 IFCHR$(VPEEK(K+J))="&"THEN41730
41710 D$(PD)=D$(PD)+CHR$(VPEEK(K+J))
41720 GOTO41690
41730 NEXTK,JJ
41740 AA=0:FORI=1TOM3:AA=AA+LC(I):NEXT
41750 IF AA>128 THEN T$="LONG. CAMPOS > 128":GOTO 41820
41760 GOTO 41850
41770 REM -----
41780 REM error
41790 REM -----
41800 T$="ERROR EN LA LINEA"+STR$(YY-1):GOTO 41820
41810 T$="CAMPOS O DESCRIP.>"+STR$(M3)
41820 LOCATE1,23:PRINT T$;" :CORREGIR ";:PLAY"O2CCC"
41830 FOR I=1 TO 4000:NEXT:LOCATE0,23:PRINTSPACE$(39);
41840 X=1:Y=YY:GOSUB41150:GOTO41070
41850 REM -------------------
41860 REM DEFINICION CALCULOS
41870 REM -------------------
41880 MX=PC:NC=0
41890 FORI=1TOMX
41900 IFTP(I)=3THEN41930
41910 LOCATEXC(I),YC(I):PRINTSPACE$(LC(I))
41920 LOCATEXC(I),YC(I):PRINTSTR$(I);
41930 NEXTI
41940 LOCATE0,23:PRINTSPACE$(39)
41950 FORI=1TOMX:FO$(I)=""
41960 IFTP(I)<>2THEN42070
41970 LOCATE0,22:PRINT"INTROD.LA FORMULA DEL CAMPO";I;"(desp.F5)";:Y=23:X=0
41980 LOCATE0,23:PRINTSPACE$(39);:PLAY"O5L32CBCB"
41990 GOSUB41310:IFI$=CHR$(29)THENX=X-1:IFX<0THENX=0
42000 IFI$=CHR$(28)THENX=X+1:IFX>30THENX=30
42010 IFI$=CHR$(20)THEN42050
42020 IFASC(I$)<32ORASC(I$)>122THEN41990
42030 PRINTI$;:X=X+1:IFX>30THENX=30
42040 GOTO41990
42050 FORH=920TO959
42060 FO$(I)=FO$(I)+CHR$(VPEEK(H)):NEXT
42070 NEXTI
42080 FORI=1TO1000:NEXT:W5=2:RETURN
42500 REM ---------
42510 REM GRABACION
42520 REM ---------
42530 CLS:LOCATE 3,8:PRINT"Rebobinar la cinta DEFINICIONES":LOCATE9,11:PRINT"y apretar REC y PLA
Y.":LOCATE7,20:PRINT"Despues pulsar una tecla."
42540 IFINKEY$=""THEN42540
42550 CLS:LOCATE 13,11:PRINT"E S P E R A R"
42560 OPEN"CAS:AZ"FOROUTPUTAS#1
42570 A$="":FORI=1TO6:PRINT#1,A$:NEXT:CLOSE#1:REM espacio cinta
42580 OPEN"CAS:DEFC"FOROUTPUTAS#1
42590 FORI=1TOM3
42600 R$=SPACE$(26)
42610 L=LEN(D$(I)):IFL<10THEND$(I)=D$(I)+SPACE$(10-L)ELSED$(I)=LEFT$(D$(I),10)
42620 XD$=STR$(XD(I)):IFLEN(XD$)=2THENXD$="0"+RIGHT$(XD$,1)ELSEXD$=RIGHT$(XD$,2)
42630 MID$(R$,1,2)=XD$
42640 MID$(R$,3,2)=RIGHT$(STR$(YD(I)),2)
```

```
42650 MID$(R$,5,10)=D$(I)
42660 MID$(R$,15,2)=RIGHT$(STR$(XC(I)),2)
42670 MID$(R$,17,2)=RIGHT$(STR$(YC(I)),2)
42680 MID$(R$,19,2)=RIGHT$(STR$(LC(I)),2)
42690 MID$(R$,21,2)=RIGHT$(STR$(TP(I)),2)
42700 MID$(R$,23,2)=RIGHT$(STR$(ND(I)),2)
42710 PRINT#1,R$:NEXTI
42720 CLOSE#1
42730 OPEN"CAS:FORC"FOROUTPUTAS#1
42740 FORI=1TOM3
42750 IFI<10THENZ$="0"+RIGHT$(STR$(I),1)ELSEZ$=RIGHT$(STR$(I),2)
42760 L=LEN(FO$(I)):IFL<30THENFO$(I)=FO$(I)+SPACE$(30-L)ELSEFO$(I)=LEFT$(FO$(I),30)
42770 R$=Z$+FO$(I)+"  "
42780 PRINT#1,R$
42790 NEXTI:CLOSE#1
42800 OPEN"CAS:PINC"FOROUTPUTAS#1
42810 PRINT#1,II$:CLOSE#1
42820 OPEN"CAS:PDFC"FOROUTPUTAS#1
42830 FORI=1TOM3:ST$=ST$+FT$(I)+NS$(I)
42840 NEXTI:PRINT#1,ST$:CLOSE#1
42850 RETURN
43000 REM --------------
43010 REM VUELVE AL MAIN
43020 REM --------------
43030 CLS:LOCATE4,10:PRINT"Rebobinar la cinta PROGRAMAS"
43040 LOCATE23,11:PRINT"=========";
43050 LOCATE12,13:PRINT"y apretar PLAY."
43060 LOCATE9,20:PRINT"Despues pulsar una tecla"
43070 IFINKEY$=""THEN43070
43080 CLEAR200:CLS:LOCATE13,11:PRINT"E S P E R A R"
43090 CLOAD"MAIN"
43500 REM -------------
43510 REM EXPLICACIONES
43520 REM -------------
43530 CLS:LOCATE1,0:PRINTH1$
43540 LOCATE 10,2:PRINT"┌──────────────────┐"
43550 LOCATE 10,3:PRINT"| CREACION MASCARA |"
43560 LOCATE 10,4:PRINT"└──────────────────┘"
43570 LOCATE3,6:PRINT"ES ESENCIAL SABER QUE:"
43580 LOCATE0,8:PRINT"1) Las descripciones  deben  estar  com-  prendidas entre 2 caracteres '&'
:p.e.   &NOMBRE&."
43590 LOCATE0,11:PRINT"2) Los caracteres  'X','N','C' en mayus-  cula definen el tipo de campo:
        X=alfanumerico, N=numerico, C=calculo."
43600 LOCATE0,15:PRINT"3) El numero de caracteres  'X','N','C'   define la longitud del campo:
   p.e.  XXXXX = campo alfanum. longitud 5."
43610 LOCATE0,19:PRINT"4) El punto entre los caracteres 'N'o'C'  separa los decimales: p.e. CCC.
CC =      campo de calculo longitud 5, de ellos   2 decimales."
43620 LOCATE10,23:PRINT"** PULSAR UNA TECLA **";
43630 IFINKEY$=""THEN43630
43640 FORI=8TO22:LOCATE0,I:PRINTSPACE$(40);:NEXT
43650 LOCATE0,8:PRINT"5) La longitud  total de  los campos, o   sea la suma de todas las 'X',las
'N' y  de las 'C', deben ser 128 como maximo"
43660 LOCATE0,12:PRINT"6) La longitud de las descripciones no    deben ser superior a 10"
43670 LOCATE0,15:PRINT"7) En las formulas de calculo el  numero  del campo debe estar entre comi
llas:     p.e. 2*(";CHR$(34);"3";CHR$(34);"+SQR(";CHR$(34);"5";CHR$(34);"))+7.5";
43680 LOCATE0,19:PRINT"8) Las funciones utilizables en los cal-  culos son:    ABS(n) ; INT(n) ;
COS(n);  SIN(n) ; SQR(n) ; TAN(n)."
43690 :IFINKEY$=""THEN43690
43700 RETURN
44000 REM -----
44010 REM AYUDA
44020 REM -----
44030 PLAY"O7L32CDE"
44040 LOCATE0,23:PRINT"  BARRA ESPACIADORA PARA CONTINUAR     ";
44050 GOSUB44180
44060 LOCATE0,23:PRINT"  F2: PARA LA INSERCION DE UN CARACTER ";
44070 GOSUB44180
44080 LOCATE0,23:PRINT"  F3: BORRA UN CARACTER                ";
44090 GOSUB44180
44100 LOCATE0,23:PRINT"  RETURN: VA AL PRINCIPIO              ";
44110 GOSUB44180
44120 LOCATE0,23:PRINT"  F5: CONTROLA, CONVALIDA Y GRABA      ";
44130 GOSUB44180
44140 LOCATE0,23:PRINT"  FLECHAS: MUEVEN EN LAS 4 DIRECCIONES ";
44150 GOSUB44180
44160 LOCATE0,23:PRINTSPACE$(39);
44170 RETURN
44180 IFINKEY$<>" "THEN44180
44190 RETURN
44500 REM ---------------------
44510 REM lectura datos mascara
44520 REM ---------------------
44530 CLS:LOCATE 3,8:PRINT"Rebobinar la cinta  DEFINICIONES":LOCATE11,11:PRINT"y apretar PLAY.":
LOCATE9,20:PRINT"Despues pulsar una tecla."
44540 IFINKEY$=""THEN44540
44550 CLS:LOCATE 14,11:PRINT"E S P E R A R"
44580 OPEN"CAS:DEFC"FORINPUTAS#1
44590 FORI=1TOM3
44600 INPUT#1,R$
44630 XD(I)=VAL(MID$(R$,1,2))
44640 YD(I)=VAL(MID$(R$,3,2))
44650 D$(I)=MID$(R$,5,10)
44660 XC(I)=VAL(MID$(R$,15,2))
44670 YC(I)=VAL(MID$(R$,17,2))
44680 LC(I)=VAL(MID$(R$,19,2))
44690 TP(I)=VAL(MID$(R$,21,2))
44700 ND(I)=VAL(MID$(R$,23,2))
44710 NEXTI:CLOSE#1
44730 OPEN"CAS:FORC"FORINPUTAS#1
44740 FORI=1TOM3
44750 INPUT#1,R$
44770 FO$(I)=MID$(R$,3,30)
44790 NEXTI:CLOSE#1
44800 RETURN
45000 REM ----------------------
45010 REM DEFINICIONES IMPRESION
45020 REM ----------------------
45025 IFW5=0THENW5=2:GOSUB44500
45030 CLS:LOCATE0,0:PRINTH1$:LOCATE1,6:PRINT"INTRODUCIR LA CABECERA (max.80 car.)"
45040 LOCATE1,8:INPUTI$:A=LEN(I$):IFA>80THEN45030
45050 IFA=80THENXP=0:GOTO45090
45060 LOCATE1,13:PRINT"INTRODUCIR LA COLUMNA DE PRINCIPIO IMPRESION"
45070 LOCATE10,15:PRINT"( min 0 ; max";80-A;")"
45080 LOCATE10,18:INPUTXP:IFXP<0ORXP>80-ATHEN45030
45090 IFXP<10THENA$="0"+RIGHT$(STR$(XP),1)ELSEA$=RIGHT$(STR$(XP),2)
45100 II$=A$+I$+SPACE$(80-LEN(I$))
45110 REM ------------------------
45120 REM SELECCION TIPO IMPRESION
45130 REM ------------------------
45140 CLS:LOCATE0,0:PRINTH1$:LOCATE5,8:PRINT"ELEGIR EL TIPO IMPRESION:"
45150 LOCATE11,12:PRINT"1) horizontal"
45160 LOCATE11,14:PRINT"2) vertical"
45170 A$=INKEY$:IFA$<>"1"ANDA$<>"2"THEN45170
45180 ST$="0"+A$
45190 CLS:LOCATE0,0:PRINTH1$
45200 LOCATE2,5:PRINT"EN LA PROXIMA MASCARA DEBERAN      ";CHR$(31);"     SELECCIONARSE LOS CAMPO
S A IMPRIMIR"
45210 LOCATE2,9:PRINT"INSERTAR LA LETRA  'T' SI EL CAMPO";CHR$(31);"      (numerico o de calculo)
 DEBE SER";CHR$(31);"      TOTALIZADO."
45220 LOCATE0,17:PRINT"p.e.:      2-T1-15-T6  =impresion de los campos 2,1,15,6,con totalizacion
de los  campos 1 y 6"
45230 LOCATE10,23:PRINT"PULSAR UNA TECLA ";
45240 IFINKEY$=""THEN45240
45250 CLS:GOSUB41020:FORI=1TOM3:IFLEFT$(D$(I),3)="   "THEN45270
45260 LOCATEXD(I),YD(I):PRINTD$(I)
45270 NEXT
45280 FORI=1TOM3:IFTP(I)=0THEN45300
45290 LOCATEXC(I),YC(I):PRINTSTR$(I)
45300 NEXT
45310 LOCATE0,23:PRINT"INDICA CAMPOS A IMPRIMIR (p.e. 2-T3-1)"
45320 LOCATE0,23:INPUTB$:LL=LEN(B$)
45330 A$="":C=0:FORI=1TOLL
45340 C$=MID$(B$,I,1):IFC$="-"THENGOSUB45370:GOTO45360
45350 A$=A$+C$
45360 NEXTI:I=I-1
45370 REM
45380 C=C+1
45390 L=LEN(A$):IFLEFT$(A$,1)="T"THENFT$(C)="01":L=L-1:A$=MID$(A$,2,L)ELSEFT$(C)="00"
45400 V=VAL(A$):IFV=0THEN 45470:REM ERROR
45410 IF TP(V)=0 THEN 45470
45420 IF TP(V)=3 THEN FT$(C)="00"
45430 IF L>2 OR L<1 THEN 45470
45440 IF L=1 THEN NS$(C)="0"+A$ ELSE NS$(C)=A$
45450 IF I=LL THEN 45560
45460 A$="":RETURN
45470 REM --- error ---
45480 CLS:LOCATE14,11:PRINT"E R R O R":PLAY"O3L16CCCC"
45490 FOR I=1 TO 1200:NEXT:FOR I=1 TO M3:FT$(I)="":NS$(I)="":NEXT
```

```
45500 CLS:LOCATE10,10:PRINT"1) CONTINUO"
45510 LOCATE10,12:PRINT"2) VUELVO AL MENU"
45520 LOCATE13,17:PRINT"CUAL ?"
45530 A$=INKEY$:IFA$<>"1"ANDA$<>"2"THEN45530
45540 IFA$="2"THENRETURN
45550 GOTO 45250
45560 FOR I=C+1 TO M3
45570 FT$(I)="00":NS$(I)="00":NEXT
45580 REM --- control longitud ---
45590 IFST$="02"THEN45640
45600 B=1:FORI=1TOM3:A=VAL(NS$(I)):IFA=0THEN45620
45610 B=B+LC(A)+1
45620 NEXTI
45630 IF B>80 THEN 45470
45640 RETURN
46000 REM ----
46010 REM MENU
46020 REM ----
46030 CLS:LOCATE0,0:PRINT H1$
46040 LOCATE11,5:PRINT"MENU DEFINICIONES"
46050 LOCATE11,6:PRINT"—————————————— "
46060 LOCATE5,11:PRINT"1) creacion mascara"
46070 LOCATE5,13:PRINT"2) definicion formatos impresion"
46080 LOCATE5,15:PRINT"3) grabacion ficheros definiciones"
46090 LOCATE5,17:PRINT"4) retorno al menu principal"
46100 LOCATE16,21:PRINT"CUAL ?"
46110 A$=INKEY$:IF A$<>"1" AND A$<>"2" AND A$<>"3" AND A$<>"4" THEN LOCATE23,21,1:FOR I=1 TO 300
:NEXT:LOCATE23,21,0:FOR I=1 TO 300:NEXT:GOTO 46110
46120 PRINT A$:FOR I=1 TO 1000:NEXT
46130 OP Z=VAL(A$)
46140 RETURN
```

Main

```
10    REM -------------
20    REM BASE DE DATOS
30    REM -------------
32    CLS:CLEAR 8700:DEFINTA-Y
35    KEY OFF:KEY 1,CHR$(15):KEY 2,CHR$(16):KEY 3,CHR$(17):KEY 4,CHR$(19):KEY 5,CHR$(20):WIDTH 4
0
37    M3=20:MR=120
40    DIM TF(15),TP(M3),LC(M3),ND(M3),CX(M3),CY(M3),DX(M3),DY(M3),HTP(M3),HLN(M3),HND(M3),IS(MR)
,NS(M3),FT(M3),Z8(M3)
50    DIM B$(M3),D$(M3),F$(M3),DA$(MR)
70    A1=1:A2=31:BP$="V1008L32CDE":NS$="#":AS$="-":PD$=".":BL$=" "
75    F$=CHR$(1)+CHR$(87)
77    O$="":FOR I=1 TO 38:O$=O$+F$: NEXT I
100   US$=CHR$(13):ZE$=CHR$(48):GIU$=CHR$(32)
110   HE$="CTRL Y (AYUDA)"
115   H1$="E.G.S.            Ediciones Forum, S.A."
120   REM CARGA VOCES
130   RESTORE
140   READ PROG$
150   READ VN
160   FOR I=1 TO VN:READ VO$(I):NEXT
170   MA=VN+1
180   VO$(MA)="Fin de tarea"
190   FOR I=1 TO 15
200   READ TF(I)
210   NEXT I
212   FORI=1TO5:READOP$(I):NEXT
215   FORI=1TO6:READFZ$(I):NEXT
219   REM -------
220   REM M A I N
221   REM -------
230   GOSUB 1300:REM MENU
235   CLS:COLOR15,4
240   IF OPZ=0 THEN FOR I=1 TO 1000: NEXT:CLS:COLOR 15,4 :END
250   ON OPZ GOSUB12000,4500,5500,14000,12000,16000
290   GOTO 230
600   REM --- ESPERAR ----
610  CLS:X=13:Y=12:GOSUB11000:PRINT"E S P E R A R":RETURN
700   REM ---------------
703   REM POSICIONA CINTA
705   REM ---------------
710   CLS:PLAYBP$:T1$="Rebobinar la cinta "+N$:X1=(40-LEN(T1$))/2:T2$="y apretar "+T$+".":X2=(40
-LEN(T2$))/2
720   X=X1:Y=9:GOSUB11000:PRINTT1$
730   X=X2:Y=13:GOSUB11000:PRINTT2$:X=9:Y=20:GOSUB11000:PRINT"y despues pulsar una tecla."
750   IF INKEY$ =""THEN 750
760   RETURN
1300  REM --------------
1310  REM MENU PRINCIPAL
1320  REM --------------
1322  CLS:COLOR15,5
1335  X=1:Y=0:GOSUB 11000:PRINT H1$
1340  L=LEN(PROG$):LL=(40-L)/2
1345  X=LL:Y=3:GOSUB11000:PRINTPROG$;
1347  X=LL-1:Y=4:GOSUB11000:FORI=1TOL+2:PRINTCHR$(&HF9);:NEXT
1350  FORI=1TOVN+1:X=4:A=I:IFI=VN+1THENA=0
1360  Y=5+I*2:GOSUB11000:PRINT".";STR$(A);" /  ";VO$(I);:NEXT
1370  LOCATE25,23:PRINT"CUAL ? .   /";
1380  I$=INKEY$:IFI$=""THENLOCATE33,23,1:FORI=1TO500:NEXT:PLAY"V406L64C":LOCATE35,23,1:FORI=1TO5
00:NEXT:PLAY"E":GOTO1380
1390  OPZ=VAL(I$):IFOPZ<=VNANDOPZ>0ORI$="0"THEN1400ELSE1380
1400  LOCATE33,23,0:PRINTOPZ;:PLAYBP$
1410  FORI=1TO1000:NEXT:RETURN
2000  REM ----------------
2010  REM PRESENTA MASCARA
2020  REM ----------------
2030  NUC=M3:CLS:COLOR15,4
2130 L=LEN(VO$(OPZ))
2140 FM=2:FM$=SPACE$(FM)
2150 X=1:Y=0:GOSUB11000:PRINTH1$;
2160 X=0:Y=1:GOSUB11000:PRINTCHR$(1);CHR$(&H58);+O$+CHR$(1);CHR$(&H59);
2170 FORY=2TO21:GOSUB11000:PRINT"|"+SPACE$(38)+"|":NEXTY
2180 X=0:Y=22:GOSUB11000:PRINTCHR$(1);CHR$(&H5A)+O$+CHR$(1);CHR$(&H5B);
2219 Y=23:X=25:GOSUB 11000
2220 PRINT HE$;
2240 FOR J=1 TO NUC
2242 IFD$(J)=SPACE$(10)THEN2272
2270 X=DX(J):Y=DY(J):GOSUB 11000:PRINT D$(J)
2272 NEXTJ
2273 FOR J=1 TO NUC :A$=""
2274 IFTP(J)=0THEN2420
2275 IFTP(J)<>5 AND TP(J)<> 4 THEN2290
2280 X=CX(J):Y=CY(J):GOSUB11000
2285 PRINTB$(J):GOTO2420
2290 N1=LC(J):N2=ND(J):IF TP(J)=1 OR TP(J) = 2 THEN C$=NS$:GOTO 2310
2300 C$=AS$
2310 A$=STRING$(N1-N2-1,C$)
2320 IF N2=0 THEN A$= A$+C$:GOTO 2350
2330 A$=A$+PD$
2340 FOR I=1 TO N2:A$=A$+C$:NEXT I
2350 X=CX(J):Y=CY(J):GOSUB 11000
2360 PRINT A$
2380 B$(J)= SPACE$ (N1-N2-1)
2410 B$(J)= B$(J)+RIGHT$ (A$,N2+1)
2420 NEXT J
2430 RETURN
3000 REM ----------
3010 REM DATA ENTRY
3020 REM ----------
3030 J=1:NUC=M3
3040 NC=J:X=CX(NC):Y=CY(NC)
3050 CC=1
3060 GOSUB 3240
3070 IF FS=0 THEN 3100
3080 IF FS>4 THEN 3210
3090 GOTO 3120
3100 J=J+1:IF J>NUC THEN 3030
3110 GOTO 3040
3120 ON FS GOTO 3130,3150,3170,3190
3130 IF J>1 THEN J=J-1:GOTO 3040
3140 PLAY BP$ :FS=0:J=NUC:GOTO 3040
3150 IF J< NUC THEN J=J+1:GOTO 3040
3160 PLAY  BP$:FS=0:GOTO 3030
3170 IF CC<LC(J) THEN S=1:CC=CC+1:X=X+1:GOSUB 11000:GOTO 3060
3180 PLAY  BP$:FS=0:GOTO 3100
3190 IF CC>1 THEN S=-1:CC=CC-1:X=X-1:GOSUB 11000:GOTO 3060
3200 PLAY  BP$:FS=0:GOTO 3040
3210 IF FS>8 THEN FS=0:GOTO 3060
3220 IF FS=8 THEN FS=0:GOTO 3100
3230 RETURN
3240 REM -------------------
3250 REM LECTURA DE UN CAMPO
3260 REM -------------------
3270 FS=0
```

```
3275 IF TP(NC)=0 THEN RETURN
3280 IF TP(NC)=4 THEN RETURN
3290 IF TP(NC)=2 THEN GOSUB8500:RETURN
3300 IF MID$(B$(NC),CC,1)=PD$ THEN X=X+S:CC=CC+S
3310 GOSUB 11000
3320 A$=INKEY$:IFA$=","OR A$= "" THEN  PRINT"█";:GOSUB11000:PRINTMID$(B$(NC),CC,1);:GOSUB11000:G
OTO3320
3330 CA=ASC(A$): IF CA > A2 THEN 3380
3340 FOR I=1 TO 15
3350 IF TF(I)= CA  THEN FS = I: GOSUB 3630: RETURN
3360 NEXT I
3370 GOTO 3320
3380 IF TP(NC)<>1 THEN 3440
3390 IF CA=45 OR CA=46 OR (CA)=48 AND CA<=57) THEN 3410
3400 PLAY BP$:GOTO 3320
3410 IF CA <> 46 THEN 3440
3420 GOSUB 4270
3430 IF SWP=1 THEN 3400
3440 S=1: GOSUB 3520
3450 IF CC=LC(NC) THEN RETURN
3460 CC=CC+1:X=X+1:GOTO 3300
3520 REM -----------------------
3530 REM TRANSFERENCIA AL BUFFER
3540 REM -----------------------
3550 IF CC=1 THEN SN$= "":GOTO 3570
3560 SN$=LEFT$(B$(NC),CC-1)
3570 IF CC=LC(NC) THEN DS$= "":GOTO 3590
3580 N=LC(NC)-CC:DS$=RIGHT$ (B$(NC),N)
3590 B$(NC)=SN$+A$+DS$
3620 PRINT A$:RETURN
3630 REM --------
3640 REM COMANDOS
3650 REM --------
3660 IF FS>4 THEN 3690
3680 RETURN
3690 IF FS=9 THEN GOSUB 13000: RETURN
3700 IF FS=8 THEN GOSUB 4080:RETURN
3710 IF FS<8 THEN RETURN
3720 ON FS-9 GOSUB 3740,3910
3730 RETURN
3740 REM ---------
3750 REM INSERCION
3760 REM ---------
3770 IF CC=LC(NC) THEN PLAY BP$:RETURN
3780 IF CC=1 THEN SN$= "":GOTO 3800
3790 SN$=LEFT$(B$(NC),CC-1)
3800 N=LC(NC)-CC+1:DS$=RIGHT$ (B$(NC),N)
3810 IF TP(NC)=1 OR TP(NC)=2 THEN C$=BL$:GOTO 3830
3820 C$=BL$
3830 A$=SN$ +C$+DS$
3840 B$(NC)=LEFT$(A$,LC(NC))
3850 XX=X:YY=Y
3860 X=CX(NC):Y=CY(NC):GOSUB 11000
3870 X=XX:Y=YY
3880 PRINT B$(NC):S=-1
3890 IF RIGHT$(B$(NC),1)= PD$ THEN 3780
3900 RETURN
3910 REM -------
3920 REM BORRADO
3930 REM -------
3940 IF TP(NC)<>1  THEN 3960
3950 IF MID$ (B$(NC),2,1) = PD$ AND CC=1 THEN RETURN
3960 IF CC<1 THEN PRINT BP$: RETURN
3970 IF CC=LC(NC) THEN DS$= "": GOTO 4000
3980 N=LC(NC) -CC:DS$ =RIGHT$ (B$(NC),N)
3990 IF CC=1 THEN SN$= "": GOTO 4010
4000 SN$=LEFT$  (B$(NC),CC-1)
4010 IF TP(NC) = 1 OR TP(NC)=2 THEN C$=NS$:GOTO 4030
4020 C$=A$$
4030 B$(NC)= SN$ +DS$ +C$
4040 XX=X:YY=Y
4050 X=CX(NC):Y=CY(NC):GOSUB 11000
4060 X=XX:Y=YY
4070 PRINT B$(NC);:S=1:RETURN
4080 REM ---------
4090 REM ALINEADOS
4100 REM ---------
4110 IF LC(NC) =1 THEN RETURN
4120 N1=LC(NC) -CC+1
4130 IF CC=1 THEN B$(NC) = "": GOTO 4150
4140 B$(NC)= LEFT$ (B$(NC),CC-1)
4150 IF TP(NC) =1 OR TP(NC)=2 THEN 4200
4160 FOR I=1 TO N1
4170 B$(NC)=B$(NC)+BL$
4180 NEXT I
4190 GOTO 4240
4200 FORI=1 TO N1
4210 B$(NC)=BL$ +B$(NC)
4220 NEXT I
4230 IF RIGHT$ (B$(NC),1) =PD$ THEN B$(NC)= BL$ +LEFT$ (B$(NC),LC(NC)-1)
4240 X=CX(NC):Y=CY(NC)
4250 GOSUB 11000
4260 PRINT B$(NC):RETURN
4270 REM ----------
4280 REM CTRL PUNTO
4290 REM ----------
4300 SWP = 0
4310 IF CC=1 OR CC=LC(NC) THEN SWP =1:RETURN
4320 FOR I=1 TO LC(NC)
4330 IF MID$(B$(NC),I,1)= PD$ THEN I = LC(NC): SWP=1: RETURN
4340 NEXT I: RETURN
4500 REM ------------------
4510 REM INTRODUCCION DATOS
4520 REM ------------------
4530 GOSUB7000
4550 CLS:X=3:Y=13:GOSUB11000:PRINT"1) EL FICHERO EXISTE":Y=15:GOSUB11000:PRINT"2) EL FICHERO DEB
E CREARSE":X=10:Y=18:GOSUB11000:PRINT"Cual ? _"
4560 A$=INKEY$:IFA$<>"1"ANDA$<>"2"THEN4560
4570 X=X+7:GOSUB11000:PRINTA$:FORI=1TO400:NEXT
4580 IFA$="1"THENGOSUB5200ELSEMX=0:GOTO4650
4600 MX=VAL(DA$(0))
4650 REM - PRINCIPIO BUCLE INTRODUCCION
4655 ON ERRORGOTO30000
4660 FORKJ=MX+1TOMR:H=KJ
4670 GOSUB2000:X=2:Y=23:GOSUB11000:PRINT"REGISTRO n.";H;:GOSUB3000
4680 IFFS=6THEN4670ELSEIFFS=7THENRETURN
4690 DA$(KJ)="*"
4700 FORI=1TOM3:IFTP(I)=0THEN4720
4710 DA$(KJ)=DA$(KJ)+B$(I)
4720 NEXTI
4730 DA$(0)=STR$(KJ)
4740 CLS:X=6:Y=11:GOSUB11000:PRINT"OTRA INTRODUCCION ? (S o N)
4750 A$=INKEY$:IFA$="N"ORA$="n"THEN4780
4760 IFA$<>"s"ANDA$<>"S"THEN4750
4770 NEXTKJ:KJ=KJ-1
4780 GOSUB5000:RETURN
5000 REM ---------------
5010 REM GRABACION DATOS
5020 REM ---------------
5030 N$="DATOS":T$="REC y PLAY":GOSUB700
5035 CLS:X=1:Y=9:GOSUB11000:INPUT"NOMBRE DEL FICHERO (max.6 car.)";NF$
5037 GOSUB 600
5038 ON ERROR GOTO 5090
5040 OPEN"CAS:AZ"FOR OUTPUT AS#1
5042 A$="":FORI=1TO6:PRINT#1,A$:NEXT:CLOSE#1:REM espacio cinta
5045 OPEN"CAS:"+NF$ FOR OUTPUT AS#1
5050 FORI=0TOVAL(DA$(0))
5060 PRINT#1,DA$(I):NEXTI:CLOSE#1
5065 IFQ$=""THENRETURN
5070 ONERRORGOTO0:Q6=2:RETURN
5090 CLS:X=9:Y=11:GOSUB11000:PRINT"Nombre fichero no valido":PLAYBP$:FORI=1TO1300:NEXT:RESUME503
0
5200 REM -------------
5210 REM LECTURA DATOS
5220 REM -------------
5225 IFQ6<>2THENQ6=2ELSERETURN
5230 N$="DATOS":T$="PLAY":GOSUB700
5235 CLS:X=1:Y=9:GOSUB11000:INPUT"NOMBRE DEL FICHERO (max.6 car.)";NF$
5237 GOSUB 600
5238 ON ERROR GOTO 5290
5240 OPEN"CAS:"+NF$ FOR INPUT AS#1
5245 INPUT#1,DA$(0)
5250 FORI=1TOVAL(DA$(0))
5260 INPUT#1,DA$(I)
5270 NEXTI:CLOSE#1:ONERRORGOTO0:RETURN
5290 CLS:X=7:Y=11:GOSUB11000:PRINT"Nombre fichero no valido":PLAYBP$:FORI=1TO1000:NEXT:RESUME523
5
5500 REM --------
```

```
5510 REM ORDENADO
5520 REM --------
5540 GOSUB7000:GOSUB5200
5550 FORI=1TOM3
5560 HTP(I)=TP(I):HLC(I)=LC(I):HND(I)=ND(I)
5570 IFTP(I)<>0THENTP(I)=3:LC(I)=1:ND(I)=0
5580 NEXTI
5590 GOSUB2000:X=1:Y=23:GOSUB11000:PRINT"'*' EN EL CAMPO CLAVE DEL ORDENADO ";:PLAYBP$+BP$:FK=2:
GOSUB3000:FK=0
5600 IFFS=6THEN5590
5605 FORI=1TOM3:TP(I)=HTP(I):ND(I)=HND(I):LC(I)=HLC(I):NEXTI
5608 IFFS=7THENRETURN
5610 FORI=1 TO M3:IFB$(I)="*"THENK4=I:GOTO5630
5620 NEXTI:CLS:X=2:Y=10:GOSUB11000:PRINT"EL CAMPO CLAVE NO SE HA INDICADO"
5625 X=12:Y=13:GOSUB11000:PRINT"VUELVO AL MENU.":PLAYBP$:FORI=1TO2500:NEXT:RETURN
5630 IFK4=1THENP1=2:P2=LC(1):GOTO5670
5640 P1=1:P2=0:FORI=1TOK4-1
5650 P1=P1+LC(I):NEXTI
5660 P1=P1+1:P2=LC(K4)
5670 GOSUB600:GOSUB6000:W6=2:P8=P1:P9=P2
5680 CLS:X=10:Y=11:GOSUB11000:PRINT"ORDENADO EFECTUADO":PLAYBP$:FORI=1TO2500:NEXT
5690 RETURN
6000 REM ----
6002 REM SORT
6005 REM ----
6010 N=VAL(DA$(0))
6020 FORI=1TON:IS(I)=I:NEXTI
6040 I1=1:J1=N
6050 I=I1:J=J1:S=-1
6060 GOSUB6220
6065 IFTP(K4)<3THENIFVAL(A1$)<=VAL(B1$)THEN6100ELSE6080
6070 IFA1$<=B1$THEN6100
6080 V=IS(I):IS(I)=IS(J)
6090 IS(J)=V:S=SGN(-S)
6100 IFS=1THENI=I+1:GOTO6120
6110 J=J-1
6120 IFI<JTHEN6060
6130 IFI+1>J1THEN6160
6140 P=P+1
6150 S9(P,1)=I+1:S9(P,2)=J1
6160 J1=I-1
6170 IFI1<J1THEN6050
6180 IFP=0THEN6210
6190 I1=S9(P,1):J1=S9(P,2):P=P-1
6200 GOTO6050
6210 RETURN:REM --- FIN ORDENADO ---
6220 A1$=MID$(DA$(IS(I)),P1,P2)
6230 B1$=MID$(DA$(IS(J)),P1,P2)
6240 RETURN
7000 REM ---------------------
7010 REM LECTURA DATOS MASCARA
7020 REM ---------------------
7025 IFO8=2THENRETURNELSEO8=2
7030 N$="DEFINICIONES":T$="PLAY":GOSUB700:GOSUB600
7040 OPEN"CAS:DEFC"FORINPUTAS#1
7050 FORI=1TOM3
7060 INPUT#1,A$
7070 DX(I)=VAL(LEFT$(A$,2)):DY(I)=VAL(MID$(A$,3,2)):D$(I)=MID$(A$,5,10)
7080 CX(I)=VAL(MID$(A$,15,2)):CY(I)=VAL(MID$(A$,17,2))
7090 LC(I)=VAL(MID$(A$,19,2)):TP(I)=VAL(MID$(A$,21,2)):ND(I)=VAL(MID$(A$,23,2))
7100 NEXTI:CLOSE#1
7110 OPEN"CAS:FORC"FORINPUTAS#1
7120 FORI=1TOM3
7130 INPUT#1,A$
7140 F$(I)=MID$(A$,3,30)
7150 NEXTI:CLOSE#1
7160 OPEN"CAS:PINC"FORINPUTAS#1
7170 INPUT#1,A$:CLOSE#1
7180 XP=VAL(LEFT$(A$,2)):II$=MID$(A$,3,80)
7190 OPEN"CAS:PDFC"FORINPUTAS#1
7200 INPUT#1,A$:CLOSE#1
7210 FI=VAL(LEFT$(A$,2))
7220 C1=3:FORI=1TOM3
7230 FT(I)=VAL(MID$(A$,C1,2)):NS(I)=VAL(MID$(A$,C1+2,2)):C1=C1+4
7240 NEXT:A$=""
7245 L=LEN(II$):IFL=0THEN7270
7250 FORI=LEN(II$)TO1STEP-1
7255 IFMID$(II$,I,1)<>" "THEN7265
7260 L=L-1:NEXTI
7265 II$=LEFT$(II$,L)
7270 RETURN
8500 REM ---------------
8501 REM PREPARA FORMULA
8502 REM ---------------
8520 A$=F$(NC)
8530 L=LEN(A$):FORI=1TOL
8540 IFMID$(A$,I,1)=CHR$(34)THEN8560
8550 NEXTI:GOTO8620
8560 A1$=MID$(A$,I+1,1):IFMID$(A$,I+2,1)<>CHR$(34)THENA1$=A1$+MID$(A$,I+2,1):P=4ELSEP=3
8570 A=VAL(A1$):L1=LEN(B$(A)):P1=1:FORJJ=1TOL1
8580 IFMID$(B$(A),JJ,1)=" "THENP1=P1+1
8590 NEXTJJ:A2$=MID$(B$(A),P1,40):IFVAL(A2$)=0THENA2$="0"
8600 A$=LEFT$(A$,I-1)+A2$+MID$(A$,I+P,L)
8610 GOTO8530
8620 GOSUB9000
8625 IFER<>0THENB$(NC)="$"+SPACE$(LC(NC)-1):GOSUB4240:PLAY"O2L2AA":J=0:RETURN
8630 L=LEN(A$):PP=1
8640 FORK1=1TOL:IFMID$(A$,K1,1)="."THEN8657
8650 NEXTK1:A1$=A$:A2$=".":GOTO8660
8657 A1$=LEFT$(A$,K1-1):IFND(NC)<>0THENA2$=MID$(A$,K1,ND(NC)+1)
8660 IFND(NC)=0THEN A2$="":PP=0:GOTO8670
8665 IF(LEN(A2$)-1)<ND(NC)THENA2$=A2$+STRING$(ND(NC)-LEN(A2$)+1,"0")
8670 N=LC(NC)-ND(NC)-PP:IFLEN(A1$)>NTHENB$(NC)="$"+SPACE$(LC(NC)-1):GOSUB4240:PLAY"O3L2C":J=0:RE
TURN
8680 B$(NC)=A1$+A2$:L=LEN(B$(NC))
8700 B$(NC)=SPACE$(LC(NC)-L)+B$(NC):GOSUB4240:RETURN
9000 REM -------
9010 REM CALCULO
9020 REM -------
9030 REM
9040 L2=LEN(A$):C$="":FORI=1TOL2:CH$=MID$(A$,I,1)
9050 IFCH$<>" "THENC$=C$+CH$
9060 NEXTI:A$=C$:C$=""
9070 GOTO9350
9080 '  *CAL 2*
9090 ER=0:L=LEN(CA$):IFL<3THENRETURN
9100 FORI=2TOL-1:CH$=MID$(CA$,I,1):FORJJ=1TO5:IFOP$(JJ)=CH$THEN9120
9110 NEXTJJ,I:ER=1:RETURN
9120 CH$=MID$(CA$,I+1,1):IFCH$="*"ORCH$="^"ORCH$="/"THENER=2:RETURN
9130 Z1=VAL(MID$(CA$,1,I-1)):Z2=VAL(MID$(CA$,I+1,40)):ONJJGOSUB9150,9160,9180,9190,9200:CA$=STR$
(Z1):IFZ1>=0THENCA$="+"+MID$(CA$,2,40)
9140 RETURN
9150 Z1=Z1^Z2:RETURN
9160 IFZ2=0THENER=3:RETURN
9170 Z1=Z1/Z2:RETURN
9180 Z1=Z1*Z2:RETURN
9190 Z1=Z1-Z2:RETURN
9200 Z1=Z1+Z2:RETURN
9210 REM --- FORM 2 ---
9220 ER=0
9230 FOROP=1TO5:LO=LEN(FO$):KK=2:IFLO=1THENKK=1
9240 PO=0:P1=0:P2=0:FORIO=KKTOLO-1:CH$=MID$(FO$,IO,1):CH=ASC(CH$):IFCH$=OP$(OP)ANDPO=0THENPO=IO:
GOTO9280
9250 O=0:IF(CH=94ORCH<48)ANDCH<>46THENO=1
9260 IFO=1ANDPO=0THENP1=IO
9270 IFO=1ANDPO>0ANDIO-PO>1THENP2=IO:GOTO9300
9280 NEXTIO:IFPO>0THEN9300
9290 NEXTOP:RETURN
9300 IFP1=0THENP1=1:GOTO9320
9310 CH$=MID$(FO$,P1,1):IFCH$<>"+"ANDCH$<>"-"THENP1=P1+1
9320 P2=P2-1:IFP2<0THENP2=LO
9330 CA$=MID$(FO$,P1,P2-P1+1):GOSUB9090:IFER<>0THENRETURN
9340 GOSUB9640:FO$=MID$(FO$,1,P1-1)+CA$+MID$(FO$,P2+1,40):GOTO9220
9350 REM --- PAR 1 ---
9360 ER=0:LA=LEN(A$):IFLA<3THEN9410
9370 PI=0:PF=0:FORIA=1TOLA:CH$=MID$(A$,IA,1):IFCH$="("THENPI=IA
9380 IFCH$=")"THEN9450
9390 NEXTIA:IFPI>0THENER=4:RETURN
9400 FO$=A$:GOSUB9210:A$=FO$
9410 IFVAL(A$)=0THENA$="0"
9420 IFMID$(A$,2,1)="."THENA$=LEFT$(A$,1)+"0"+MID$(A$,2,80)
9430 IFLEFT$(A$,1)="+"THENA$=MID$(A$,2,80)
9440 RETURN
9450 IFPI=0THENER=4:RETURN
9460 PF=IA:FO$=MID$(A$,PI+1,PF-PI-1):GOSUB9210:IFER<>0THENRETURN
9470 IFPI>3THENGOSUB9520
9480 IFPI<=1THEN9500
9490 IFA<0ANDMID$(A$,PI-1,1)="-"THENPI=PI-1:FO$="+"+MID$(FO$,2,80)
```

```
9500 IFER<>0THENRETURN
9510 A$=MID$(A$,1,PI-1)+FO$+MID$(A$,PF+1,80):GOTO9360
9520 REM --- FUN 1 ---
9530 ER=0:CH$=MID$(A$,PI-1,1):IFCH$<"A"ORCH$>"Z"THENRETURN
9540 FU$=MID$(A$,PI-3,3):FORI=1TO6:IFFU$=FZ$(I)THEN9560
9550 NEXTI:ER=5:RETURN
9560 ZA=VAL(FO$):ONIGOSUB9570,9580,9590,9600,9610,9630:FO$=STR$(ZA):CA$=FO$:GOSUB9640:FO$=CA$:IF
ZA>=0THENFO$=MID$(FO$,2,40):PI=PI-3:RETURNELSEPI=PI-3:RETURN
9570 ZA=ABS(ZA):RETURN
9580 ZA=INT(ZA):RETURN
9590 ZA=COS(ZA):RETURN
9600 ZA=SIN(ZA):RETURN
9610 IFZA<0THENER=6:RETURN
9620 ZA=SQR(ZA):RETURN
9630 ZA=TAN(ZA):RETURN
9640 REM --- ESP 1 ---
9650 ZO$=STRING$(64,"0"):L=LEN(CA$)
9660 FORIC=1TOL:IFMID$(CA$,IC,1)="E"THEN9680
9670 NEXT:RETURN
9680 N$=MID$(CA$,2,IC-2):L=LEN(N$)
9690 FORI=1TOL:IFMID$(N$,I,1)="."THENP=I:GOTO9710
9700 NEXT:P=L
9710 E=VAL(MID$(CA$,IC+1,3)):IFE<0THEN9790
9720 DD=L-P:IFDD=0THEN9770
9730 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80)
9740 IFDD<=ETHEN9770
9750 N$=MID$(N$,1,P+E-1)+"."+MID$(N$,P+E,80)
9760 DD=E
9770 N$=N$+MID$(ZO$,1,E-DD)
9780 GOTO9830
9790 IFP=LTHENP=P+2-L:GOTO9810
9800 N$=MID$(N$,1,P-1)+MID$(N$,P+1,80):L=1
9810 N$=MID$(ZO$,1,-E-L+1)+N$
9820 N$=MID$(N$,1,P-1)+"."+MID$(N$,P,80)
9830 CA$=LEFT$(CA$,1)+N$
9840 RETURN
10000 REM -----
10010 REM DATOS
10020 REM -----
10030 DATA "BASE DE DATOS"
10040 DATA 6
10050 DATA "definicion mascara","introduccion datos","ordenado","busqueda con variacion","defini
cion formatos impresion","impresion"
10070 DATA 30,31,28,29,20,16,15,13,25,19,17,0,0,0,0
10080 DATA"^","/","*","-","+"
10090 DATA"ABS","INT","COS","SIN","SQR","TAN"
11000 REM ----------------
11010 REM POSICIONA CURSOR
11020 REM ----------------
11030 LOCATE X,Y
11040 RETURN
12000 REM ------------------------------
12010 REM DEFINICION MASCARA E IMPRESION
12020 REM ------------------------------
12030 COLOR15,4
12040 N$="PROGRAMA":T$="PLAY":GOSUB700:GOSUB600:CLEAR200
12070 CLOAD"CREA"
12080 STOP
13000 REM -----
13010 REM AYUDA
13020 REM -----
13030 IFFK=2THENRETURN
13050 LOCATE 0,23
13055 PRINT "AYUDA(barra espaciadora para continuar)";
13060 GOSUB 13600
13110 PRINT "FLECHAS: mueven en las 4 direciones    ";
13120 GOSUB 13600
13130 PRINT "RETURN:alinea y convalida introduccion ";
13140 GOSUB 13600
13150 PRINT "F1: vuelve al menu principal           ";
13160 GOSUB 13600
13170 PRINT "F2: anula toda la introduccion         ";
13180 GOSUB 13600
13190 PRINT "F3: borra un caracter                  ";
13200 GOSUB 13600
13210 PRINT "F4: permite la insercion de espacios   ";
13220 GOSUB 13600
13222 PRINT "F5: graba toda la introduccion         ";
13224 GOSUB 13600
13230 PRINT "                        CTRL Y (AYUDA) ";
13235 LOCATE1,23:PRINT"REGISTRO n.";H;
13300 RETURN
13600 A$=INKEY$:IF A$= "" THEN 13600
13605 IF A$<> CHR$(32) THEN 13600
13610 LOCATE 0,23
13620 RETURN
14000 REM ----------------------
14010 REM BUSQUEDA CON VARIACION
14020 REM ----------------------
14030 CLS:COLOR15,8:X=0:Y=0:GOSUB11000:PRINTH1$
14040 X=9:Y=4:GOSUB11000:PRINTSTRING$(LEN(VO$(4)),"-")
14050 Y=Y+1:GOSUB11000:PRINTVO$(4)
14060 Y=Y+1:GOSUB11000:PRINTSTRING$(LEN(VO$(4)),"-")
14070 X=3:Y=12:GOSUB11000:PRINT"INTRODUCIR EL NUMERO DE REGISTRO";STRING$(2,31);"          (0 =
BUSCA EN CAMPO CLAVE)    ";CHR$(31);"          (-1 = VUELVE AL MENU)"
14080 X=16:Y=20:GOSUB11000:INPUTA$:IFLEN(A$)>3THEN14030
14090 R=VAL(A$):IFR>VAL(DA$(0))THEN14030
14100 IFR<0THENRETURN
14110 IFR<>0THENW6=0:MIS=R:MAS=R:GOTO14250
14120 IFW6<>2THENCLS:X=2:Y=11:GOSUB11000:PRINT"PARA ESTE TIPO DE BUSQUEDA, ANTES";CHR$(31);"
  HAY QUE HACER AL ORDENADO.":PLAYBP$+BP$:FORI=1TO1800:NEXT:GOTO14000
14130 X=0:FORY=10TO21:GOSUB11000:PRINTSPACE$(40);:NEXT
14140 X=2:Y=11:GOSUB11000:PRINT"INTRODUCIR EL VALOR DEL CAMPO CLAVE"
14150 X=2:Y=15:GOSUB11000:PRINT"NOTA. El campo clave debe ser el mismo      que el del ordenado.
"
14160 X=6:Y=19:GOSUB11000:PRINT:INPUTRA$
14162 IFLEN(RA$)>LC(K4)THEN14130
14165 CLS:X=8:Y=6:GOSUB11000:PRINT"ESPERAR, ESTOY BUSCANDO:":X=(40-LEN(RA$))/2:Y=8:GOSUB11000:PR
INTRA$
14170 IFTP(K4)=3THENRA$=RA$+SPACE$(LC(K4)-LEN(RA$)):GOTO14190
14180 RA$=SPACE$(LC(K4)-LEN(RA$))+RA$
14190 GOSUB14500:REM RICERCA BIN.
14195 FORI=1TO1500:NEXT
14200 X=1:Y=6:GOSUB11000:PRINTSPACE$(38):PLAYBP$
14210 IFTR=0THENX=9:Y=13:GOSUB11000:PRINT"N O   H A L L A D O":PLAYBP$:FORI=1TO2000:NEXT:GOTO140
00
14220 A=MAS-MIS+1:X=13:Y=13:GOSUB11000:PRINT"H A L L A D O":X=12:Y=19:GOSUB11000
14230 IFA=1THEN PRINT"(un solo registro)"ELSE PRINT"(";A;" registros)"
14240 PLAYBP$:FORI=1TO2500:NEXT
14250 REM --- PRESENTA REGISTRO ---
14260 FOR JK=MIS TO MAS :KJ=IS(JK)
14265 IFR<>0THENKJ=JK
14270 A$=MID$(DA$(KJ),2,LEN(DA$(KJ))-1)
14280 C1=1:FORI=1TOM3:IFTP(I)=0THEN14300
14290 B$(I)=MID$(A$,C1,LC(I)):C1=C1+LC(I):HTP(I)=TP(I):TP(I)=4
14300 NEXTI
14310 IFR=0THENH=KJELSEH=JK
14320 GOSUB2000:X=2:Y=23:GOSUB11000:PRINT"RECORD n.";H;
14330 FORI=1TOM3:TP(I)=HTP(I):NEXT:IFR=0THENTP(K4)=4
14340 GOSUB3000:TP(K4)=HTP(K4)
14350 IFFS=6THEN14270ELSEIFFS=7THENRETURN
14360 DA$(KJ)="*"
14370 FORI=1TOM3:IFTP(I)=0THEN14390
14380 DA$(KJ)=DA$(KJ)+B$(I)
14390 NEXTI
14400 X=1:Y=23:GOSUB11000:PRINT"                  PULSAR UNA TECLA          ";:PLAYBP$
14410 IFINKEY$=""THEN14410
14420 NEXTJK:CLS:X=7:Y=11:GOSUB11000:PRINT"OTRA VARIACION ? (S o N)"
14430 A$=INKEY$:IF A$="s" OR A$="S" THEN 14000
14440 IFA$<>"N"ANDA$<>"n"THEN14430
14450 GOSUB 5000:RETURN
14460 REM ----------------
14500 REM BUSQUEDA BINARIA
14510 REM ----------------
14515 P1=P8:P2=P9
14520 RI=0:RF=VAL(DA$(0)):RM=INT(RF/2)
14530 RE=IS(RM):RB$=MID$(DA$(RE),P1,P2)
14540 IFRA$=RB$THEN14640
14550 IFRA$>RB$THEN14600
14560 RF=RM:RM=INT((RF-RI)/2)
14570 IFRM<1THENRM=1
14580 RM=RM+RI:IFRM>=RFTHENTR=0:RETURN
14590 GOTO14530
14600 RI=RM:RM=INT((RF-RI)/2)
14610 IFRM<1THENRM=1
14620 RM=RM+RI:IFRM>RFTHENTR=0:RETURN
14630 GOTO14530
14640 SG=-1:RF=RM:RI=RM:MIS=RM:MAS=RM:FMI=0
```

```
14650 RM=RM+SG:RE=IS(RM):RB$=MID$(DA$(RE),P1,P2)
14660 IFRA$=RB$THEN14690
14670 IFFMI=1THENTR=1:RETURN
14680 FMI=1:SG=1:RM=RF:GOTO14650
14690 IFFMI=1THENMAS=MAS+1ELSEMIS=MIS-1
14700 GOTO14650
16000 REM ---------
16010 REM IMPRESION
16020 REM ---------
16030 IFW6<>2THENCLS:X=1:Y=9:GOSUB11000:PRINT"ANTES DE LA IMPRESION HAY QUE HACER";CHR$(31);"
 EL ORDENADO.  VUELVO AL MENU":PLAYBF$+BP$:FORI=1TO5000:NEXT:RETURN
16045 GOSUB600:GOSUB17000:CT=0
16050 FORJ=1TOM3:Z8(J)=0:NEXT
16055 FORI=1TOMR
16060 IFIS(I)=0THEN16200
16065 IFCT>35THENCT=0:LPRINTCHR$(12):GOSUB17000
16070 A$=MID$(DA$(IS(I)),2,200)
16080 C1=1:FORJ=1TOM3:IFTP(J)=0THEN16100
16090 B$(J)=MID$(A$,C1,LC(J)):C1=C1+LC(J)
16100 NEXTJ
16105 IFFI=2THENLPRINT"-------------":LPRINT"REGISTRO N.";IS(I):LPRINT"-------------"
16110 A$="!":FORJ=1TOM3:IFNS(J)=0THEN16150
16120 IFFT(J)=1THENZ8(J)=Z8(J)+VAL(B$(NS(J)))
16130 IF FI=1THENA$=A$+B$(NS(J))+"!":GOTO16150
16140 LPRINTB$(NS(J)):CT=CT+1
16150 NEXTJ
16160 IF FI=1THENLPRINTA$:CT=CT+1:GOTO16170
16165 LPRINTSTRING$(80,"-"):CT=CT+4
16170 NEXTI
16200 REM --- TOTALES ---
16210 LPRINTTAB(0);STRING$(80,"-")
16220 IFFI=2THEN16300
16230 A=1:FORI=1TOM3:IFTP(I)=0THEN16260
16240 A=A+LC(NS(I))+1:IFFT(I)=0THEN16260
16245 AA=A-1-LEN(STR$(Z8(I))):IFAA<0THENAA=0
16250 LPRINTTAB(AA);Z8(I);
16260 NEXTI:LPRINT
16280 GOTO16350
16300 LPRINTTAB(8)"T O T A L E S":LPRINTTAB(8)"***********"
16310 FORI=1TOM3:IFTP(I)=0THEN16340
16320 IFFT(I)=0THENLPRINT:GOTO16340
16330 LPRINTTAB(10);Z8(I)
16340 NEXTI
16350 LPRINTCHR$(12):RETURN
17000 REM --- CABECERA ---
17010 LPRINTSTRING$(80,"_"):LPRINT
17020 LPRINTTAB(XP);II$
17030 LPRINTTAB(0);STRING$(80,"_")
17040 RETURN
30000 CLS:REM --- error ---
30010 IFERR=14THEN30050
30020 PRINT"ERROR ";ERR;"EN LA LINEA";ERL:STOP
30050 DA$(KJ)="":DA$(KJ-1)="":DA$(KJ-2)="":KJ=KJ-3:DA$(0)=STR$(KJ):ERASEB$,D$,VO$:O$="":REM CREA
 ESPACIO
30060 X=4:Y=9:GOSUB11000:PRINT"EL ESPACIO EN MEMORIA ESTA AGOTADO";CHR$(31);" TENGO QUE BORRAR L
OS 3 ULTIMOS REGISTROS"
30070 X=1:Y=20:GOSUB11000:PRINT"PULSAR UNA TECLA PARA GRABAR EL FICHERO"
30080 IF INKEY$=""THEN30080
30090 GOSUB5000:RUN
```

RELACION DE LAS PRINCIPALES VARIABLES - PHILIPS VG 8010

Definiciones

D$(*)	= descripciones
TP(*)	= tipos de campos
LC(*)	= longitud campo de introducción
ND(*)	= número de decimales
XD(*), **YD(*)**	= coordenadas en el vídeo de las descripciones
XC(*), **YC(*)**	= coordenadas de los campos
FO$(*)	= matriz de las fórmulas de cálculo
FT$(*), **NS$(*)**	= parámetros de impresión
L1$,H1$, **L2$**	= cabecera
A	= código de la tecla pulsada
XP	= columna principio impresión
II$	= cabecera de impresión

Main

B$(*)	= matriz de los buffers (1 por campo de introducción)
LC(*)	= longitud del campo de introducción
ND(*)	= número de decimales en el campo de introducción
CX(*), **CY(*)**	= coordenadas en el vídeo del campo de introducción
DX(*), **DY(*)**	= coordenadas en el vídeo de la descripción
TF(*)	= matriz numérica que contiene los códigos ASCII
D$(*)	= descripciones en la máscara vídeo
BL$	= blank
PD$	= punto decimal
AS$	= carácter "–"; identifica los campos de introducción alfanuméricos
NS$	= carácter "#"; identifica los campos de introducción numéricos
NUC	= número de campos de introducción en la máscara vídeo activada
MR	= número máximo de registros
HT(*), **HL(*)**, **HN(*)**	= matriz auxiliar para máscara vídeo ordenado
IS(*)	= indicadores de los registros después del ordenado
NS(*), **FT(*)**	= parámetros de impresión
Z8(*)	= totales en fase de impresión
F$(*)	= matriz de las fórmulas de cálculo
DA$(*)	= matriz de los registros
S9(*, *)	= índice para el ordenado
KJ	= índice de los bucles de introducción
MX	= último registro introducido
K4	= número del campo clave del ordenado
ERR	= indicador de error
OP$	= operadores reconocidos (↑,/,*,–,+)
FZ$	= funciones reconocidas (ABS,INT, COS,SIN,SQR,TAN)

Los gráficos de ordenador

La representación gráfica seguramente es el medio más inmediato para la comprensión de un fenómeno. En los programas anteriores se ha mostrado cómo realizar cálculos (Tablero electrónico) y archivar datos (Bases de datos): en la mayoría de los casos, estas actividades terminan con el análisis de los resultados, o sea con su interpretación. En los dos programas anteriores, la manera en que se proporcionan los valores finales es la tabular, que es poco adecuada para mostrar su proceso. Evidentemente, es mucho mejor la representación gráfica, que permite una interpretación visual inmediata.

Por este motivo, todos los «paquetes» para empleo generalizado en el ámbito profesional contienen funciones para la representación gráfica de los datos. Este programa permite desarrollar gráficos o histogramas con datos contenidos en archivos, en discos o valores introducidos manualmente por el usuario y de manera que pueda ser utilizado también por separado por otros procedimientos.
Su empleo principal es para la representación de los datos contenidos en los archivos creados con la Base de Datos. En consecuencia, este programa de gráficos es su complemento indispensable.

Uso del programa

Este tipo de aplicación (programas para la generación de los gráficos) es muy sencillo desde el punto de vista del análisis, mientras que puede requerir notables esfuerzos de programación, en particular para las versiones en máquinas que no disponen de una buena gestión de la pantalla en modo gráfico.
Para ello es necesario hacer algunas precisiones correspondientes a las tres versiones previstas (MSX, Apple, C-64). La primera (MSX) ofrece un programa «ágil» y completo en cuanto dispone de instrucciones Basic para gráficos muy potentes, análogas a las previstas en los ordenadores personales.
La segunda versión (Apple) tiene buenas capacidades gráficas, pero orientadas al «dibujo» más que a mostrar el proceso de un fenómeno, puesto que en él no es posible el uso mixto de la pantalla, o sea con gráficos y caracteres. El problema puede superarse utilizando, si bien en parte, la subrutina de presentación de caracteres en modo gráfico, ya vista en otras aplicaciones.
La última versión (C-64) es la más complicada, puesto que el Basic albergado en esta máquina está completamente falto de instrucciones para los gráficos, lo cual obliga a escribir una subrutina incluso para la simple presentación de un segmento (que en las otras máquinas se obtiene con una sola instrucción).
Por tanto, los diagramas que se representan a continuación deben considerarse como una guía de principio; el desarrollo real de los programas puede deducirse en función de la versión, aunque sus potencialidades son comunes.
Como se ha indicado, en este caso, el desarrollo de un análisis no presenta ninguna dificultad, puesto que el procedimiento no necesita ni pesadas introducciones de datos ni gestiones particulares de ficheros, reduciéndose a una simple lectura de valores y a su representación.

La finalidad del procedimiento es la representación gráfica de valores. El proceso necesario se reduce a la elección de los campos a presentar y a la reducción de sus valores según un factor de escala que depende del formato de la pantalla. El último parámetro, para la definición completa del proceso, es la definición de los datos de entrada.
Resumiendo, las funciones a realizar son:

- Adquisición de los valores a presentar como gráfico
- Elección de los campos entre los admitidos y conversión de los valores según un factor de escala
- Presentación del gráfico

En la página 2510 se han resumido, en forma de diagrama, los tres puntos anteriores.
Analizando las tres funciones es evidente que el procedimiento debe implantarse de manera muy parametrizada. Efectivamente, por lo menos hay que prever tres tipos de entradas:

- de ficheros generados con la Base de Datos
- de ficheros genéricos
- de teclado

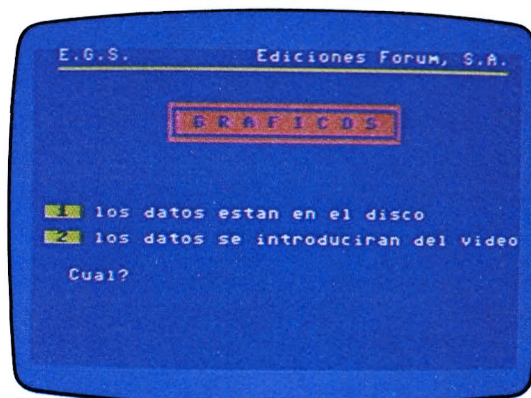
y diferentes tipos de presentación. Los activos son:

- Presentación X, Y (la forma normal utilizada para las funciones)
- Histogramas

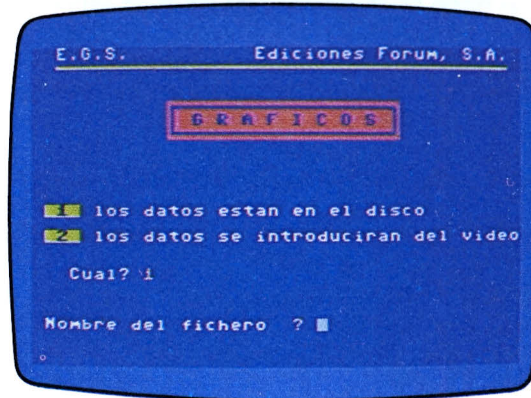
Otras formas, que pueden ser útiles en situaciones particulares, se comentan en las implantaciones.
La primera parametrización (formatos de los datos) puede obtenerse fácilmente y no necesita desarrollos particulares de programación ni presenta dificultades en su uso.

La segunda forma de presentación, además de requerir tantas subrutinas distintas como cuantas formas diferentes hay, puede generar errores de interpretación si no se utiliza correctamente.
La hipótesis con la que se ha estructurado el procedimiento prevé la carga de más campos y la elección de los que deben presentarse como gráficos.
Esta implantación es consecuencia de la finalidad primaria del procedimiento, que es la presentación gráfica de los valores contenidos en ficheros generados por la Base de Datos. En este tipo de ficheros, cada registro contiene más campos, representando cada uno una cierta magnitud, y puede ser útil disponer de un procedimiento que pueda mostrar los procesos de varias magnitudes sin necesitar, cada vez, una carga de datos.
Por ejemplo, un fichero podría contener los movimientos de cantidades de dinero de entrada y de salida, por lo que en un registro (correspondiente, pongamos por caso, a un determinado mes) puede haber tanto el importe correspondiente a los cobros como al de los pagos.
Con la estructura adoptada, los dos valores (para cada registro) se leen simultáneamente y también están simultáneamente en la memoria. Por tanto, pueden representarse gráficamente bien en modo alterno (primero uno y después otro) bien en modo conjunto, obteniendo dos formas de gráfico que son muy útiles en los análisis.
Esta técnica de análisis de los procesos no debe subvalorarse, puesto que constituye un medio muy potente para interpretar algunos fenómenos.
Supongamos que se observan dos magnitudes A y B que varían en el tiempo: el fenómeno puede representarse con las expresiones A = f(t) y B = f(t), que tienen el significado: la magnitud A es una función del tiempo (t), así como la magnitud B.
Para poner de manifiesto el proceso hay que construir dos gráficos en los que representar: en el primero los valores de A en función de los de "t" y, en el segundo, los valores de B (siempre en función de "t"). Estos dos gráficos presentan qué les sucede a las magnitudes A y B con el paso del tiempo "t", pero poco o nada sobre las eventuales relaciones entre A y B. Para analizar los dos fenómenos ajo este aspecto de interdependencia es más útil disponer de un gráfico que muestre el proceso de una magnitud en función de la otra, es decir, que muestre la relación A = f(B). Observando este gráfico puede detectarse la presencia o no de un vínculo entre A y B, o sea de una dependencia entre las dos magnitudes o, expresado en otros términos, de una «correlación».

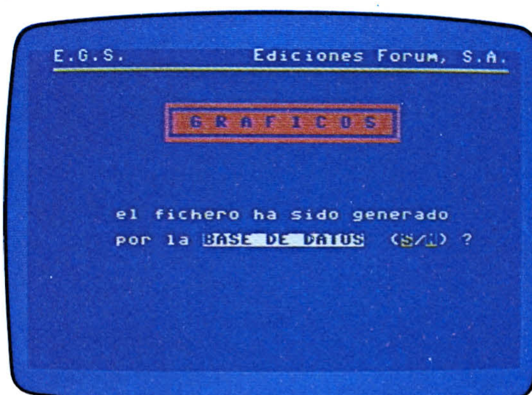
Selección del periférico utilizado para la introducción de los datos. Activando la función 1, los valores se leerán en un fichero; con la 2, deberán introducirse por teclado.



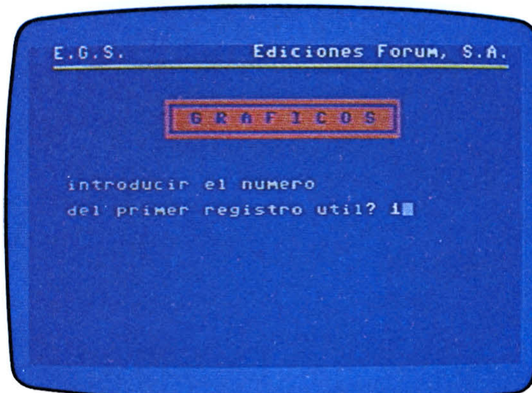
El usuario ha seleccionado la voz 1 (datos en el disco) y el sistema pide el nombre del fichero.



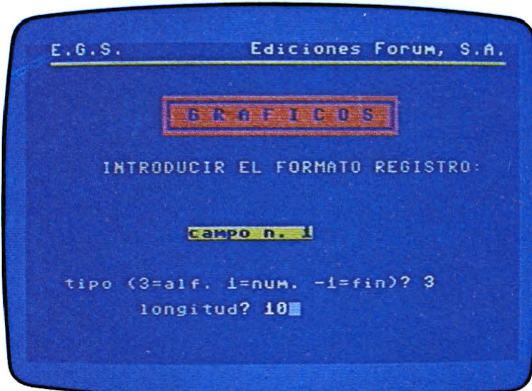
La opción 1 (datos del disco) también requiere precisar el formato de registro. En el caso de que los datos se introduzcan con la Base de Datos, es el propio sistema al que obtiene el formato a través del fichero de servicio.



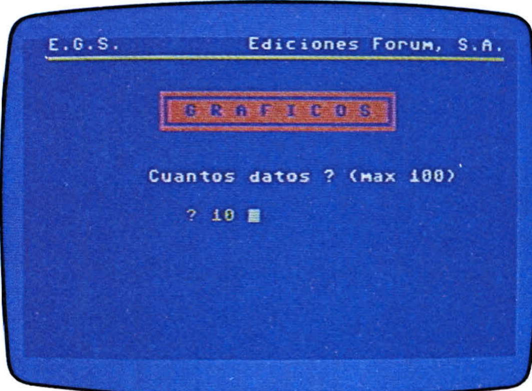
Terminadas las fases anteriores, hay que definir la parte del fichero que se va a utilizar. La foto muestra la introducción del número del registro en el que empiezan los datos.



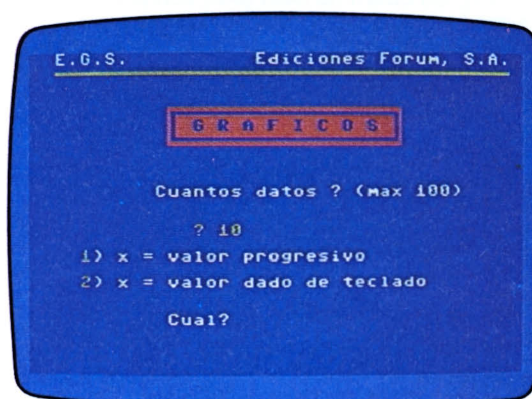
En el caso en que los datos se generan con la Base de Datos, el usuario debe definir su formato. Para cada dato hay que introducir el tipo y la longitud.



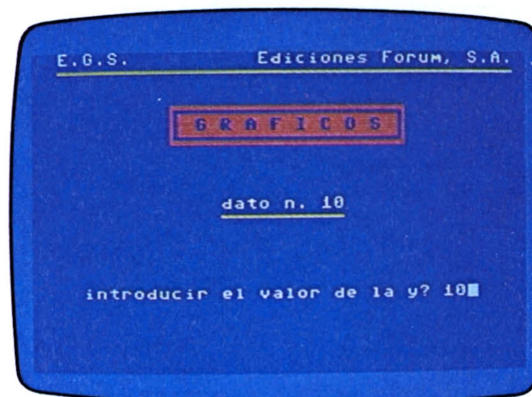
La foto muestra la máscara vídeo utilizada para definir el número de datos que se van a presentar en forma de gráfico; esta operación es necesaria sólo en el caso de valores introducidos por teclado.



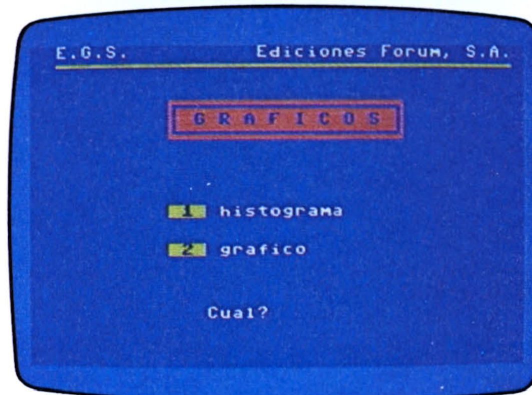
Completado de la máscara anterior, con la selección del tipo de valor a asignar al eje X.



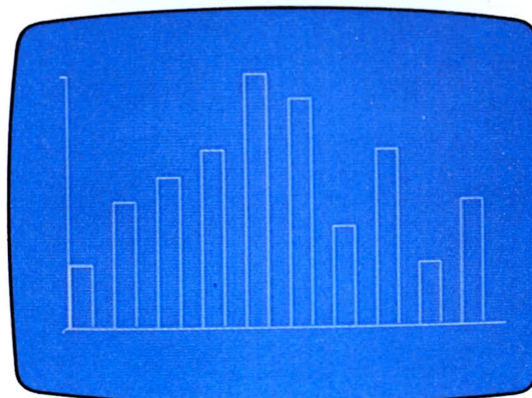
La foto muestra la fase de introducción de los valores, que sigue a las máscaras de las fotos anteriores.



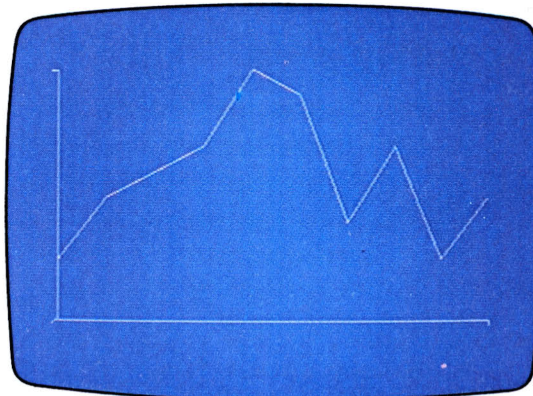
Selección del tipo de gráfico.



El usuario ha seleccionado el tipo 1 (histograma). La foto muestra su desarrollo.



Seleccionando la voz 2 se obtiene la presentación del gráfico de los valores.



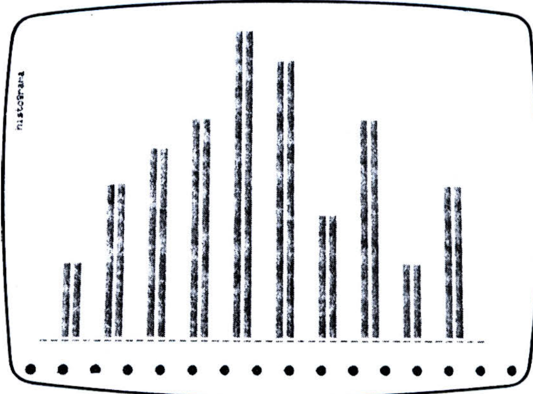
Petición del cálculo de los coeficientes de la recta de regresión: en caso de respuesta afirmativa, el sistema proporciona los coeficientes de la recta utilizando los valores del gráfico.



El histograma puede representarse sobre papel. La foto muestra el coloquio que permite la activación de esta función.



El dibujo muestra un ejemplo de histograma sobre papel obtenido con la correspondiente función.



La lógica del programa

A la luz de todo cuanto se ha expuesto en la parte dedicada al uso del programa, puede comentarse el diagrama de flujo representado abajo.

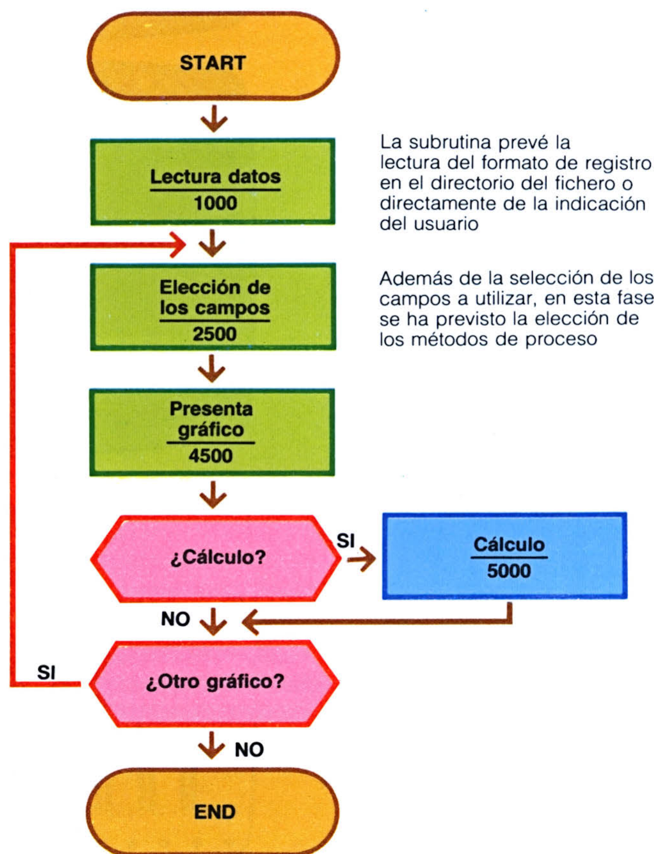
La primera función a realizar es la lectura de los datos, prevista, como se indica más adelante, de tres maneras diferentes. Siguen la elección de los campos a representar gráficamente (uno para el eje X y otro para el eje Y) y la representación del gráfico (subrutina 4500).

A la salida de esta última subrutina se ha previsto el cálculo de la recta de regresión (con el método de los mínimos cuadrados).

Al final, el sistema pregunta si se desea otro gráfico y, en caso de respuesta afirmativa, vuelve a la subrutina de elección de los campos.

Al respecto hay que hacer algunas precisiones sobre las modalidades operativas del uso de este software. Como se ha indicado, los datos se toman principalmente de los registros de los ficheros generados con la Base de Datos, lo que no implica ninguna particularidad si los valores (X e Y) están contenidos en el registro. Sin embargo, en algunos casos, una de las dos magnitudes (normalmente la correspondiente al eje X) se toma simplemente como valor progresivo. Por ejemplo, en el caso expuesto anteriormente de gestión de las entradas y de las salidas mensuales, cada registro corresponde a un mes y, al final de la memorización de los datos, sólo contiene los importes sin ninguna referencia a los meses. Efectivamente, el fichero estará compuesto por 12 registros, uno por mes, y el puntero al registro (o sea el número de registro) indica el mes correspondiente. Por tanto, a los solos fines del tratamiento de datos no es necesario memorizar por cada mes el registro al que se refiere.

Sin embargo, la estructura pensada así no es suficiente para los gráficos: de hecho no se han previsto procesos en función de variables no contenidas en el re-



gistro, lo que implica la necesidad de incluir en cada registro todas las eventuales referencias.
Por ejemplo, en el caso anterior es necesario prever una estructura de datos que en el registro incluya también el campo con la indicación del mes.
Esta particularidad funcional no constituye un elemento limitador: de hecho, puede superarse fácilmente empleando una estructura de datos adecuada.
Como alternativa podría incluirse la posibilidad de utilizar el número del registro como variable independiente, pero esta solución sólo resuelve un caso particular y se adapta poco a la estructura generalizada del procedimiento.

Lectura de datos. Como ya se ha indicado, se han previsto tres vías de entrada de datos:

1 / De un fichero generado con la Base de Datos
2 / De un fichero cualquiera
3 / Por introducción directa por parte del usuario

Los dos primeros modos son análogos y sólo se diferencian en la lógica utilizada para determinar el formato de registro. Efectivamente, para los ficheros creados con la Base de Datos, el formato se memoriza en el disco, mientras que en el otro caso debe ser especificado por el usuario.
La tercera vía de entrada, lectura del teclado, sólo se ha incluido para permitir el uso del procedimiento en los casos de empleo no ligados a un fichero datos.
Para la lectura de los valores de un fichero, el usuario debe introducir el nombre, después la selección del tipo (o sea si se ha creado o no con la Base de Datos) y la adquisición del formato. En el primer caso se llama la subrutina 2200, que utiliza la estructura generada por la Base de Datos. En el segundo se llama la subrutina 1500, a través de la cual, el usuario debe definir la estructura de registro.
Además, las dos subrutinas se utilizan para definir el número de puntos a presentar. En el caso de que el valor sea superior al permitido por la pantalla se llama la subrutina 4000, que permite realizar una agregación de los datos para limitar su número a los contenidos en la pantalla.
La parte restante de la subrutina está constituida por un bucle de lectura de datos del fichero, especificando que se realiza entre los valores IO y MX.
En el caso de ficheros generados por la Base de Datos, el valor de IO es 2, puesto que el primer registro está reservado para el directorio. Para los otros, lo debe especificar el usuario.
Las funciones realizadas pueden verse en el diagrama y no necesitan comentario, excepto una explicación del uso particular de la matriz B$(*).
En todas las aplicaciones anteriores, B$(*) se ha utilizado como buffer vídeo y, por tanto, se ha dedicado principalmente a la data entry, mientras que en este procedimiento tiene un significado completamente distinto, puesto que se emplea como matriz para la carga de los datos. De ella se extraerán sucesivamente, uno cada vez, los campos a representar gráficamente.

Lectura de datos de pantalla. La subrutina 3500 se dedica a la adquisición de los datos del teclado. El primer valor a suministrar es el número de datos (de los pares de datos, uno corresponde al eje X y el otro al eje Y) que se van a utilizar.
En esta subrutina se ha previsto la lógica de asignación con valores automáticos y progresivos al eje X (descrita anteriormente).
Sigue un bucle de 1 a MX con la petición de dos valores (X, Y) y su transferencia a las matrices X(*) e Y(*).
Naturalmente, en el caso de A$="1", o sea la asignación automática y progresiva de la X, se salta la correspondiente introducción.
En la versión presentada, los valores introducidos automáticamente empiezan por 0. De hecho, para I = 1, la asignación es X(1) = 0, pero puede desviarse o parametrizarse fácilmente planteando una pregunta al usuario.
Obsérvese que para permitir la máxima elasticidad no se han incluido controles de ningún tipo, por lo que del usuario se requiere un mínimo de atención. Por ejemplo, el usuario debe introducir los datos de manera que los valores correspondientes al eje X resulten progresivos, puesto que, de otro modo, se genera un gráfico sin significado.
Esta limitación también puede eliminarse fácilmente previendo el ordenado de los datos introducidos (sólo los correspondientes al eje X) antes de salir de la subrutina.

Lectura del formato de registro. También esta subrutina es extremadamente sencilla y no requiere ningún comentario particular.
Sólo hay que recordar que para los ficheros creados con la Base de Datos, el formato de los registros está contenido en "DEF@" (fichero de las definiciones).
Los valores se leen en la variable R$ y, de ésta, se transfieren a las matrices XD(*) e YD(*).
Las matrices X(*) e Y(*) no se utilizan directamente porque deberán contener los valores de manera adecuada para ser representados gráficamente, o sea con una cierta escala y agregados (si fuese necesario) según una determinada lógica. El uso de las matrices de tránsito XD(*) e YD(*) permite tener siempre en la memoria los valores originales y, por tanto, modificar en cualquier momento las variables presentadas.

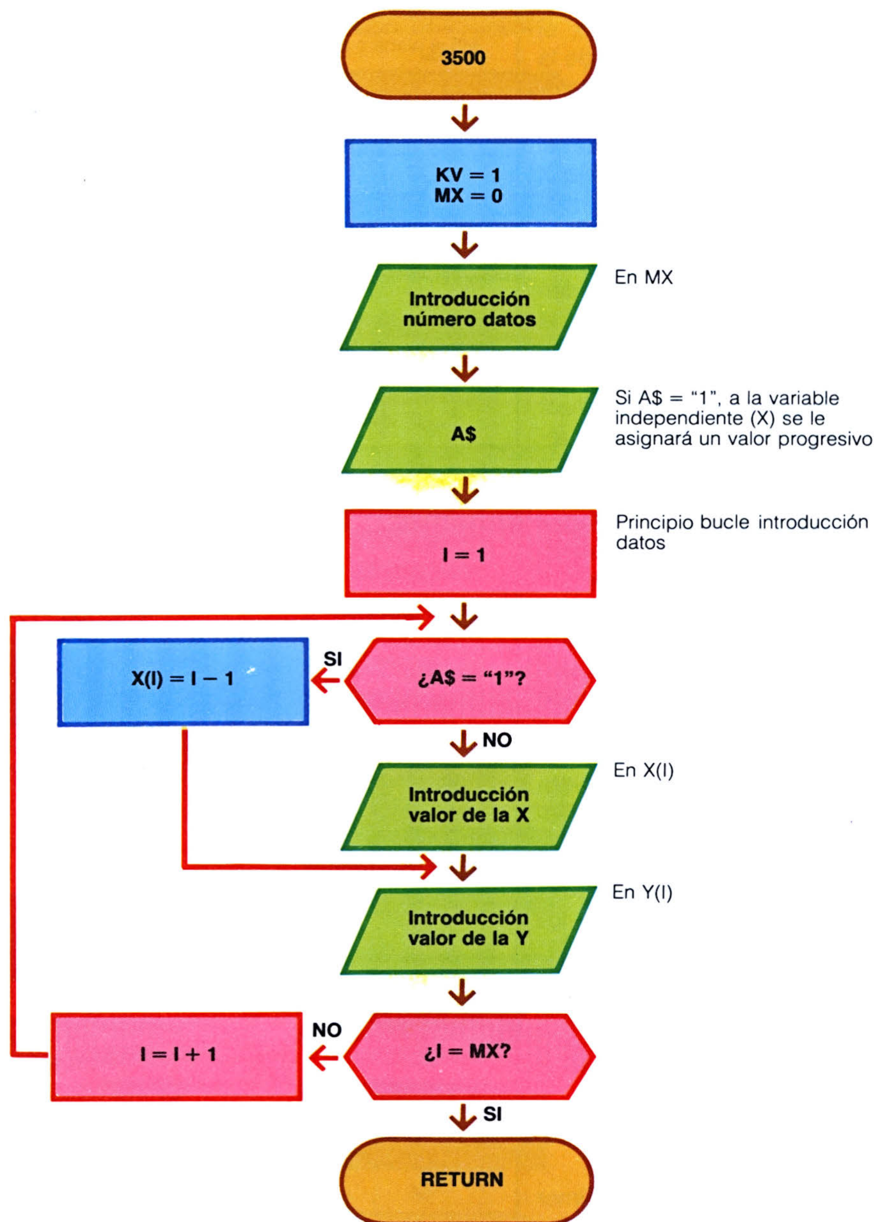
Introducción del formato de registro. Como ya se ha indicado, la subrutina se utiliza para definir el formato de registro de un fichero cualquiera del cual se desean extraer los datos. La única limitación en su empleo es que los valores hayan sido memorizados en formato ASCII, o sea en forma de cadenas.

LECTURA DE DATOS

En este caso, cada registro puede leerse en una variable de cadena y los distintos campos pueden extraerse definiendo su tipo y su longitud, como si se tratase de un fichero generado por la Base de Datos.
La primera función prevista en la subrutina es la introducción del número de registro en el que iniciar la extracción, que también podría no ser el primero del registro. Generalizando esta función también puede incluirse la introducción del último registro útil para fraccionar todo el fichero en varios grupos de datos.
Sigue un bucle con un coloquio para la introducción de los parámetros (TP(*), LC(*)) de definiciones de los campos.
Obsérvese que este bucle puede interrumpirse antes del máximo previsto (20 campos) introduciendo tipo = – 1.
A la salida se determina el número de campos utilizados (M3) y se pide al usuario el número máximo de los registros a procesar.
Sigue el cálculo de los registros entendidos como posición a lo largo del fichero y la subrutina termina.

Elección del criterio de selección. Tal como se ha indicado, pueden producirse casos en los que el número de puntos disponible sea superior al previsto en la pantalla. Por ejemplo, en la memorización de un balance (entradas y salidas), la frecuencia de las grabaciones podría ser diaria y, por tanto, se necesitarían 366 registros, valor generalmente superior al número de pixels (puntos de pantalla) previstos en el eje X (eje de los tiempos).
Esta situación conduce necesariamente a la eliminación de algunos datos en un número adecuado para reducir los restantes a la cantidad contenida en los pixels del eje X.
Obsérvese que el problema

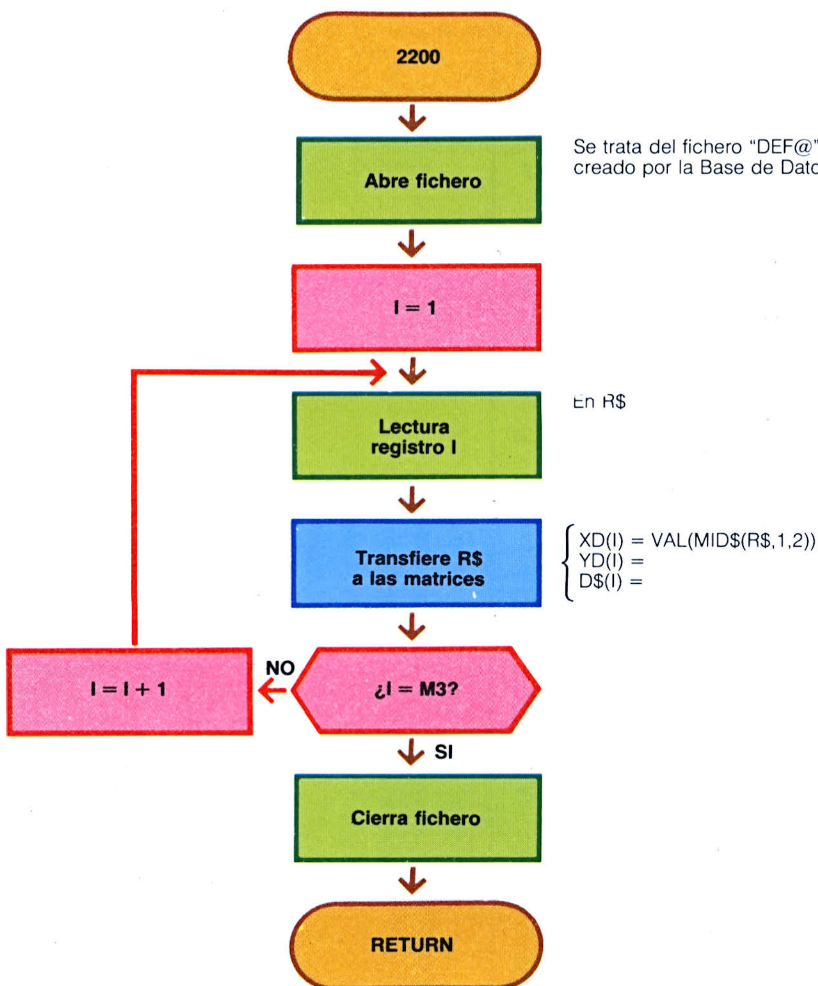
LECTURA DE DATOS EN PANTALLA



expuesto sólo se produce para el eje X, porque para el eje Y basta el cálculo de un factor de escala adecuado. El motivo de esta diferencia reside en el significado que tienen los dos ejes: el primero, eje X, es la variable independiente y cada uno de sus valores que identifica un punto del gráfico debe ser bien evidente y separado de los demás. En cambio, el eje Y contiene los valores de la magnitud que pueden representarse a escala.
Es decir, suponiendo que el número de pixels en el eje X sea de 200, los datos correspondientes al ejemplo anterior no pueden presentarse gráficamente porque corresponden a 366 días, mientras que la pantalla sólo puede contener un máximo de 200.
En este eje no se puede utilizar un factor de escala, porque reduciendo el número de días a un valor contenido en 200 pixels (por ejemplo con un factor de escala 1:2) se tiene una representación falseada y de difícil interpretación, puesto que cada punto representaría aproximadamente 1,5 días. Por tanto, hay que adoptar un método de selección de los puntos a representar gráficamente.
Los previstos en la rutina, entre los cuales el usuario debe elegir, son:

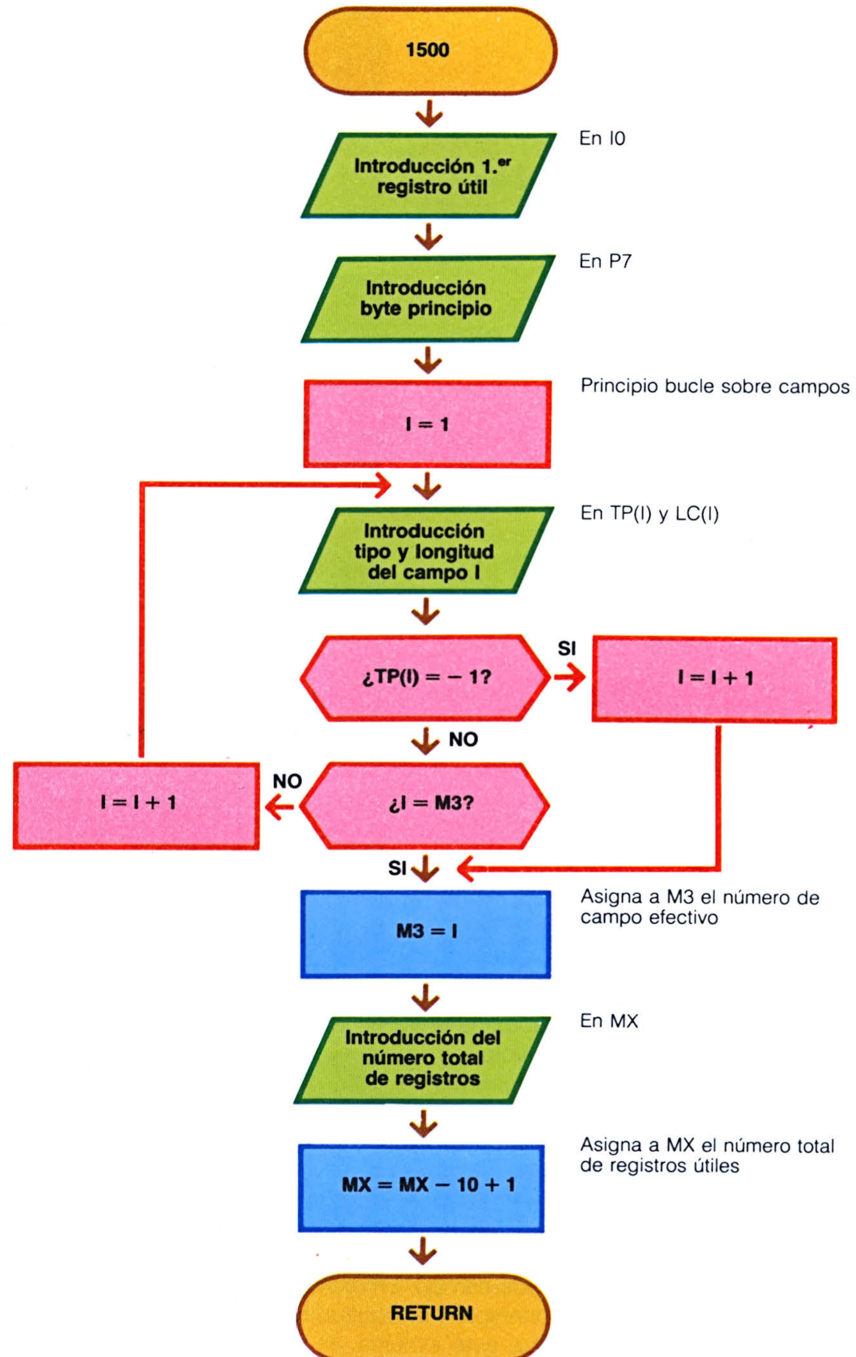
1 / Eliminación de los primeros, en un número igual a los que sobran
2 / Eliminación de los últimos, siempre en un número igual a los que sobran
3 / Presentación de un valor cada "N"
4 / Eliminación de un valor cada "N"

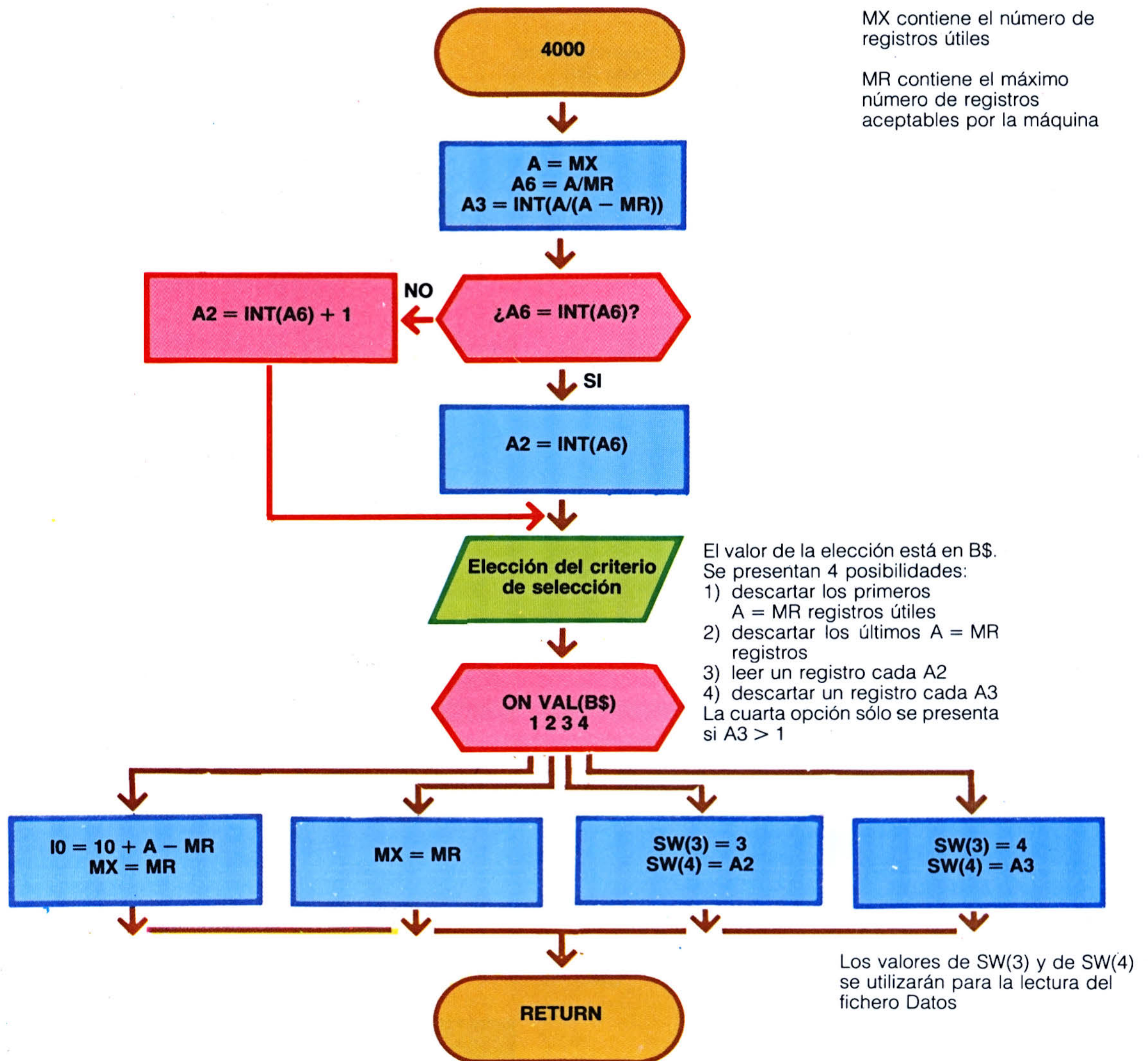
En el ejemplo anterior, el primer criterio de selección equivale a representar en el gráfico sólo los 200 primeros días, el segundo los últimos 200, el tercero a representar por ejemplo los días pares y el último crite-



rio, a eliminar en el gráfico un día de cada dos.
La elección del criterio a aplicar no puede hacerse *a priori* sin conocer los términos del problema y, por tanto, se delega al usuario, excepto en un caso que se comentará más adelante, que con la estructura actual no puede darse.
La subrutina 4000 determina si el número de puntos es superior al de los pixels y avisa al usuario, esperando instrucciones acerca del método a adoptar para su reducción (uno de los 4 antes indicados): por tanto, en la salida, los valores a presentar ya son congruentes con la estructura de la máquina y no precisan otros procesados. Para activar la lógica de selección automática (descrita más adelante) es por tanto necesario prever una salida que no permita ninguna reducción de los datos en la 4000.

Elección de los campos. La función realizada por la subrutina es la extracción, de la cadena que contiene un dato (registro), de los campos a representar gráficamente.
En las subrutinas anteriores, el procedimiento ha adquirido tanto las descripciones (en D$(*)) como los parámetros (tipo y longitud) de cada campo presente en el registro.
En esta fase (subrutina 2500), utilizando los datos anteriores, se extraen los valores (de B$(*), que contiene todos los registros) convertidos a numéricos y se memorizan en las matrices X(*) e Y(*) que se usarán para la presentación. Naturalmente, esta lógica no debe activarse en el caso de que los valores se hayan introducido directamente por teclado. Por tanto, un primer control sobre el flag KV (implantado en la llamada por la subrutina) produce la salida de la 2500 al producirse la condición KV = 1 (flag activo).





En el otro caso (KV = 0), la subrutina entra en la fase de proceso constituida por puntos.

1 / Presentación de las descripciones de los campos y elección, por parte del usuario, de los dos a emplear; el primero como variable independiente (eje X) y el otro como variable dependiente (eje Y)

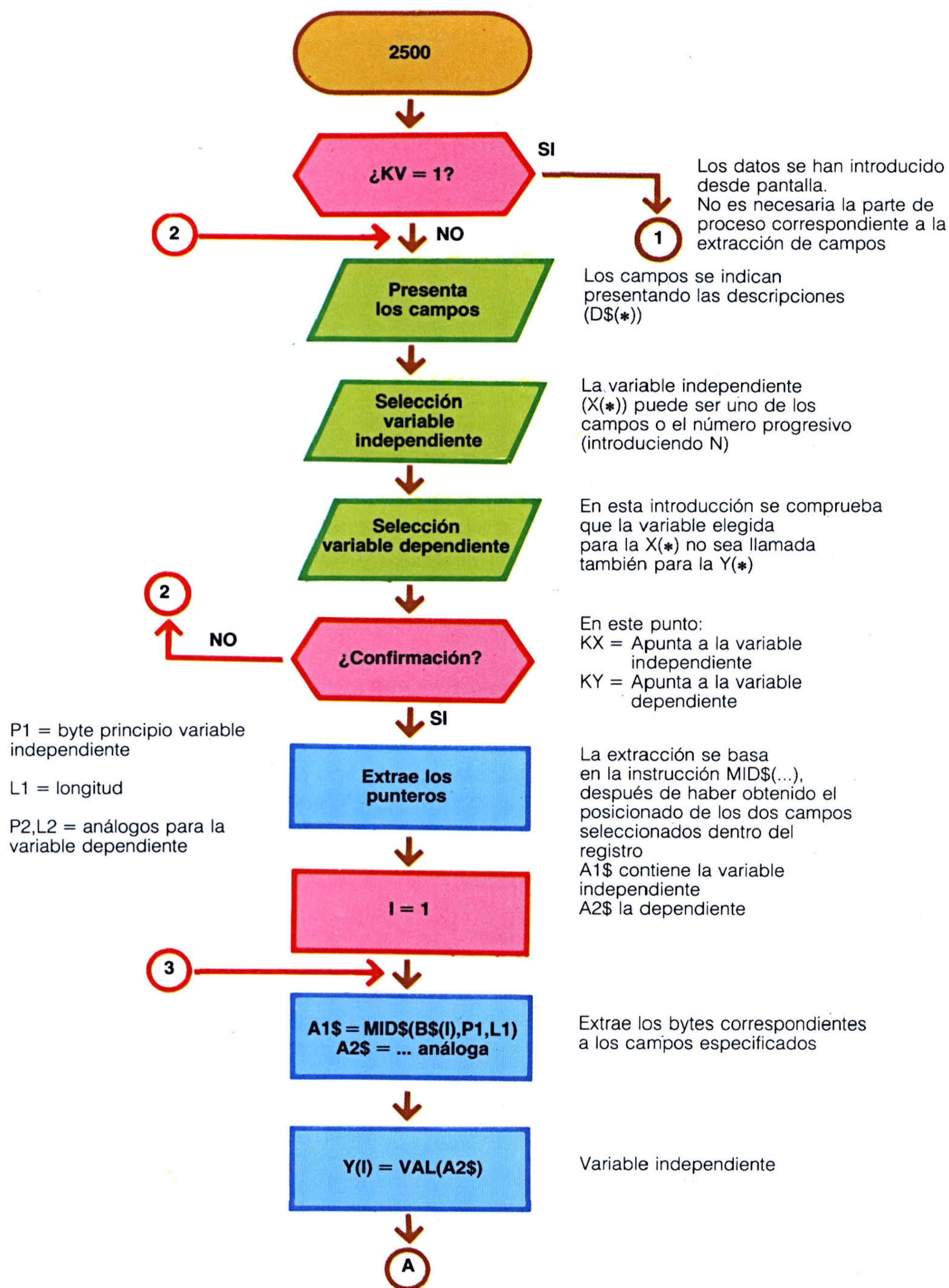
2 / Extracción de los punteros, o sea, cálculo, en base a los parámetros de definición del registro, del byte de principio y de final de cada campo. En realidad son suficientes el byte de principio y la longitud del campo (ya conocida porque está incluida en los parámetros)

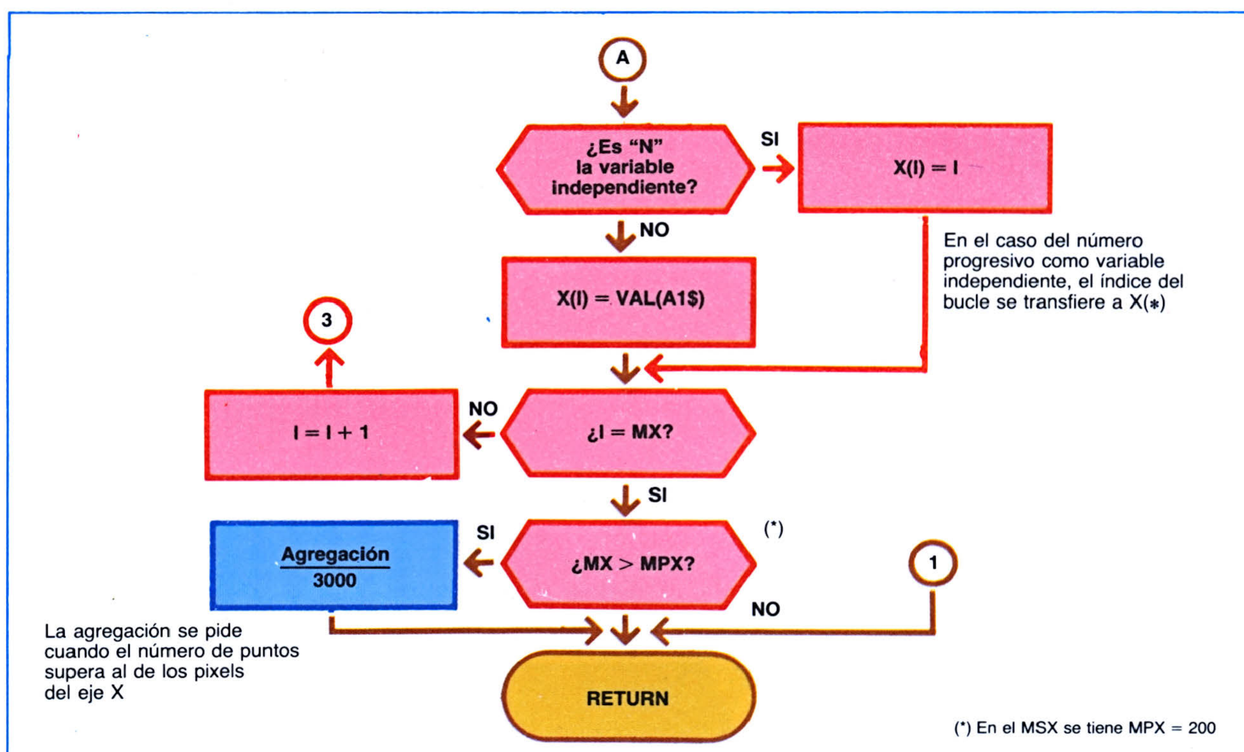
3 / Bucle de extracción de las cadenas B$(∗) de cada campo y su transferencia a X(∗) y a Y(∗).

Agregación. Como se ha indicado, en el procedimiento hay incluida una lógica automática de reducción del número de los valores a presentar.
Las consideraciones sobre las que se basa esta lógica se exponen a continuación utilizando un ejemplo.
Supongamos que se tienen 1000 valores a presentar en una pantalla que, en el eje X, tiene 200 puntos (pixels). Adoptando uno de los criterios expuestos, en la subrutina 4000 se consigue lo deseado, pero eliminando completamente algunas mediciones.
Utilizando los dos primeros métodos sólo se tiene la presentación de una parte de los valores, mientras que utilizando los otros se eliminan algunos valores que podrían ser importantes para un determinado proceso del fenómeno.

ELECCION DE LOS CAMPOS

En la llamada, KV = 1 indica que se ha hecho una introducción desde pantalla





En esta subrutina se presenta un método que, en algunos casos, resuelve el problema.
Con los valores indicados (1000 puntos, 200 pixels), se está limitado a representar un valor de la magnitud de cada cinco (1000/200 = 5) y, eligiendo este valor de manera aleatoria, puede representarse un punto momentáneo y poco significativo.
Como alternativa puede utilizarse una media «local» de los 5 valores a representar en un solo punto. O sea, en lugar de indicar una medida descartando las otras, se calcula la media de un grupo de 5, con lo que se obtiene un total de 200 valores, cada uno constituido por una media «local»
Por tanto, la subrutina está constituida por un bucle que calcula la medida de K valores (en el ejemplo anterior, K = 5) para ambas variables.
En la salida de esta subrutina se emplea un segundo bucle con el fin de anular los elementos de X(*) y de Y(*) que sobran.

Presentación del gráfico. El diagrama de esta subrutina (4500) sólo es indicativo, puesto que su desarrollo depende estrechamente del tipo de máquina utilizado. Las funciones que realiza son:

1 / Extracción de los valores máximos a presentar en los dos ejes
2 / Cálculo del factor de escala
3 / Presentación, en dos formas:
- Como gráfico X-Y
- Como histograma (subrutina 4920)

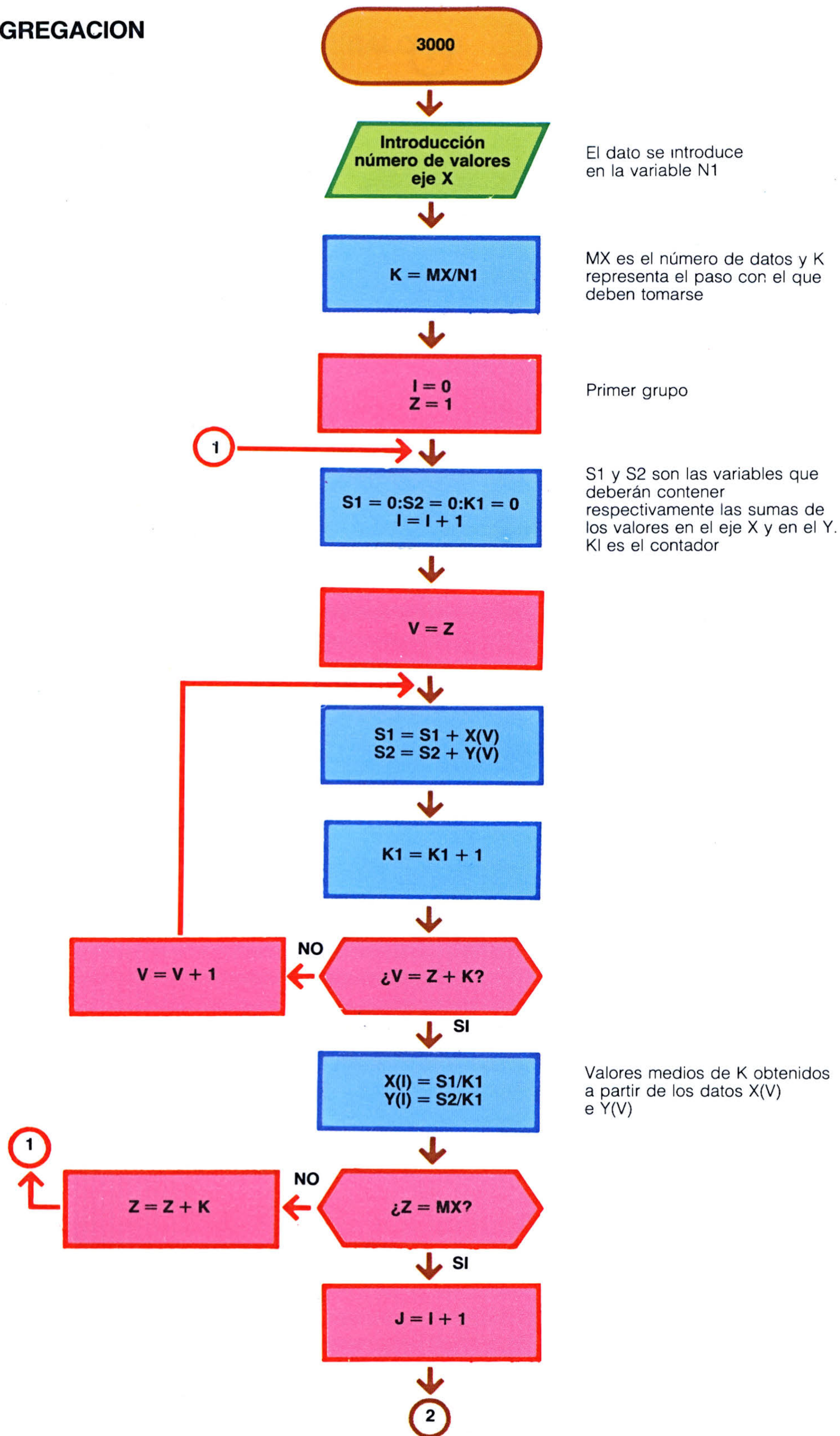
La segunda forma (histograma) puede hacerse en impresora. La subrutina tiene las siguientes limitaciones:

- No prevé valores negativos
- La representación de las variables empieza siempre por cero

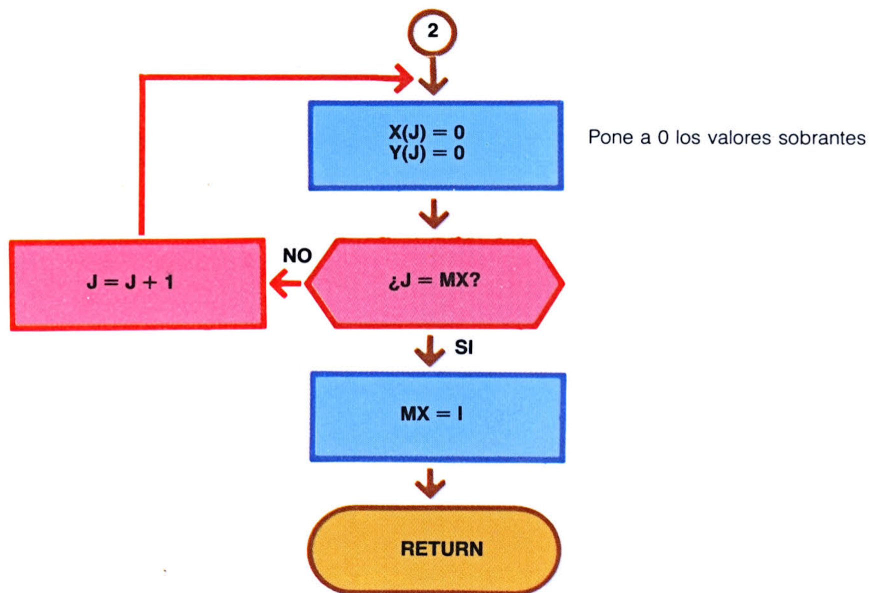
Estas limitaciones son consecuencia del sector de aplicación particular del procedimiento. De hecho, este tipo de gráfico está orientado al empleo de datos principalmente de naturaleza económica o de gestión, para lo cual dichas limitaciones no tienen un carácter determinante, como sucedería en cambio para aplicaciones de carácter científico.
Sin embargo, el propio usuario puede ampliar el campo de empleo previendo también el caso de valores negativos; el procedimiento a adoptar consiste en sumar a los valores un número correspondiente al máximo negativo (para obtener únicamente valores positivos) desplazando los ejes la misma cantidad.

Cálculo. En muchas aplicaciones, los valores a presentar gráficamente se derivan de observaciones de un fenómeno.
Cada valor, precisamente porque se obtienen mediante una observación, o sea de una medición, está afectado por errores que no pueden eliminarse.
Estos errores producen una «dispersión» de los valores observados, por lo que incluso si el fenómeno puede ser representado por ejemplo con una función matemática, no todos los puntos pertenecen a la curva,

AGREGACION

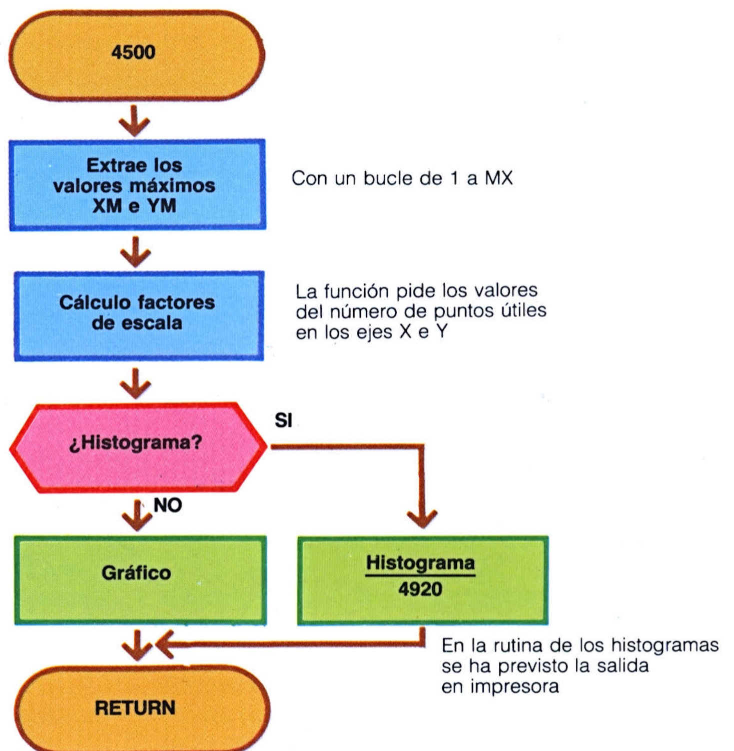


3000
Introducción número de valores eje X
El dato se introduce en la variable N1
K = MX/N1
MX es el número de datos y K representa el paso con el que deben tomarse
I = 0
Z = 1
Primer grupo
1
S1 = 0:S2 = 0:K1 = 0
I = I + 1
S1 y S2 son las variables que deberán contener respectivamente las sumas de los valores en el eje X y en el Y. KI es el contador
V = Z
S1 = S1 + X(V)
S2 = S2 + Y(V)
K1 = K1 + 1
V = V + 1
NO
¿V = Z + K?
SI
X(I) = S1/K1
Y(I) = S2/K1
Valores medios de K obtenidos a partir de los datos X(V) e Y(V)
1
Z = Z + K
NO
¿Z = MX?
SI
J = I + 1
2



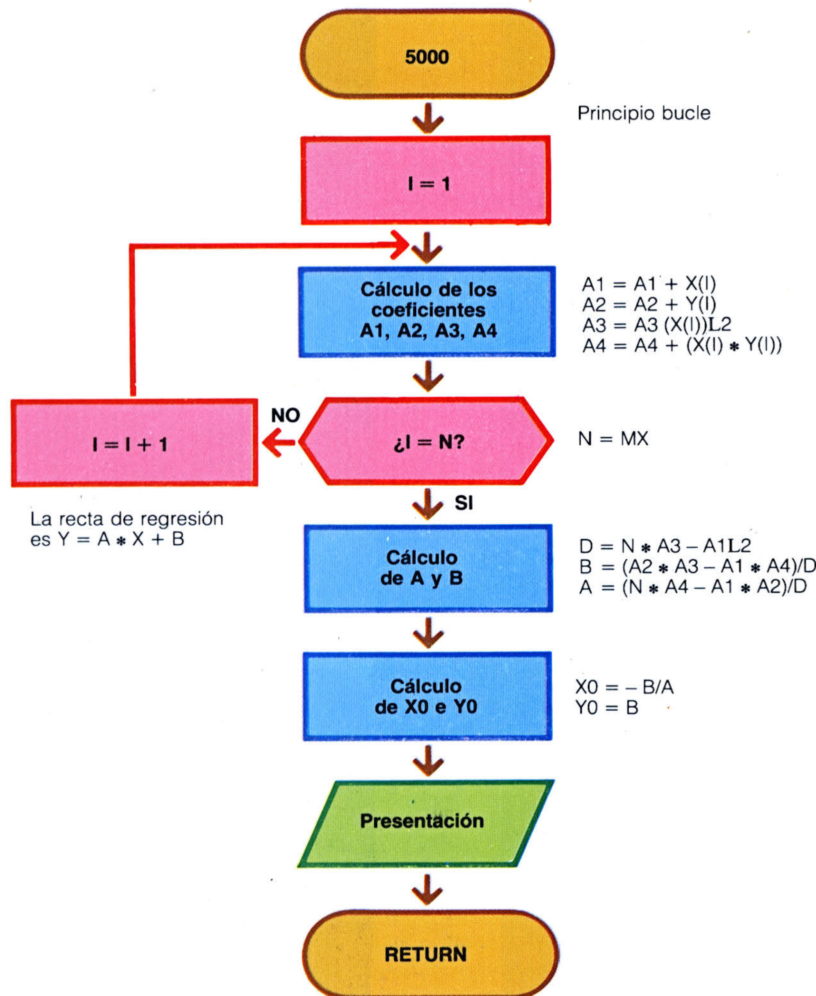
PRESENTACION DEL GRAFICO

En entrada:
X(*)= Variable independiente
Y(*)= Variable dependiente
MX = Número de puntos



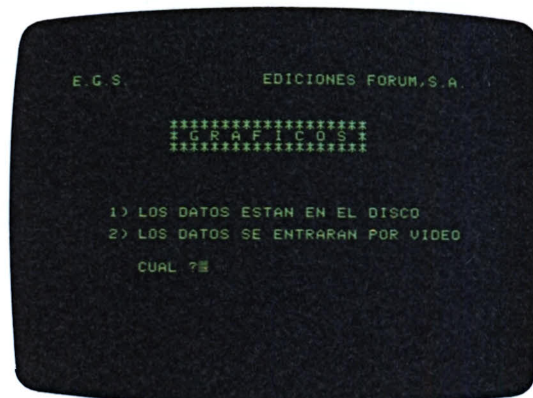
La subrutina sólo es indicativa porque su desarrollo depende del tipo de máquina utilizado. Por tanto, hay que consultar el correspondiente listado

CALCULO

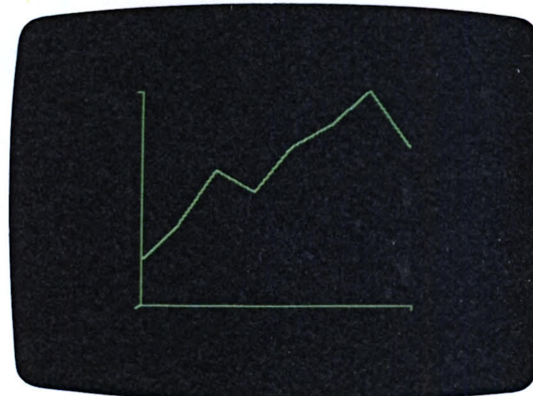


sino que están distribuidos a su alrededor, generalmente de manera aleatoria. Los métodos para determinar la distribución real son muchos y todos de una cierta complejidad, puesto que requieren conocimientos de estadística y, a veces, procesos muy complejos. Esta subrutina sólo constituye un ejemplo del método más sencillo, que presupone una relación de tipo rectilíneo entre las dos variables del gráfico.
El método adoptado es el llamado «de los mínimos cuadrados» y proporciona la recta que mejor se aproxima a la representación del fenómeno, siempre que su evolución pueda representarse con una recta.
Por tanto, esta parte del programa no constituye una forma generalizada, sino sólo la solución para un determinado tipo de fenómenos.
Sin embargo, el usuario puede construir las versiones necesarias para una aplicación particular: se trata de desarrollar una lógica distinta de cálculo que utilice los algoritmos precisos.
Este tipo de implantación no se contempla en el párrafo reservado al tema, puesto que no está ligado específicamente a los gráficos. De hecho, la presentación de los parámetros de la curva que mejor se aproxima al fenómeno presentado gráficamente, tan sólo es un cálculo matemático cuyo desarrollo no depende de ninguna manera de la parte gráfica.

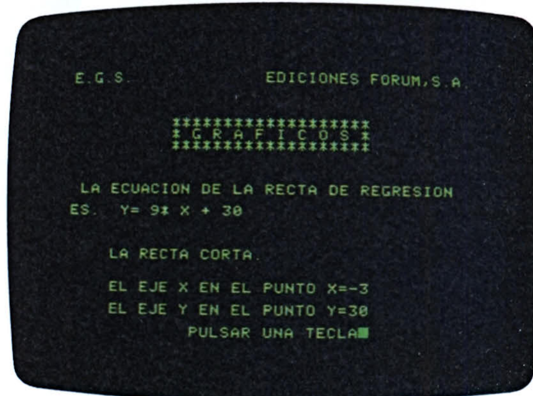
En esta foto se muestra el menú del programa de gráficos en la versión DOS. Con esta primera pantalla puede elegirse el método de inserción de los valores a procesar y presentar gráficamente: en particular, introduciendo 1 es posible leer un fichero datos cualquiera en el soporte magnético.



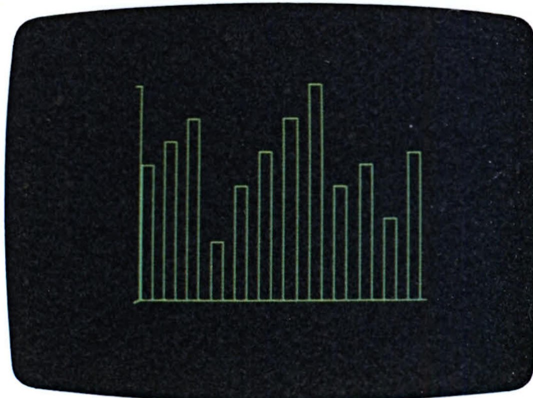
Ejemplo de gráfico: la presentación de los datos en forma de gráfico se hace por líneas, o sea cada par de valores consecutivos identifica dos puntos de la pantalla, que finalmente quedan unidos por un segmento.



Una vez terminada la presentación del gráfico, el usuario puede pedir al programa el cálculo de la recta de regresión determinada por los valores introducidos.



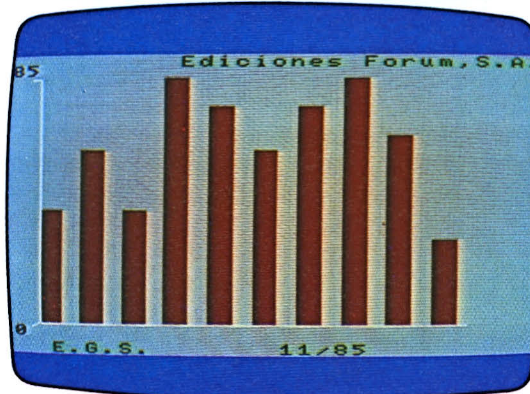
Ejemplo de histograma: la presentación de los valores procesados en forma de histograma también se hace en alta resolución. Las columnas no están llenas para no prolongar el tiempo de trazado.



El programa de gráficos en la versión MSX permite trazar histogramas o gráficos según la elección realizada en el menú secundario mostrado en esta foto. Las dos formas de presentación se realizan con pantalla de alta resolución.



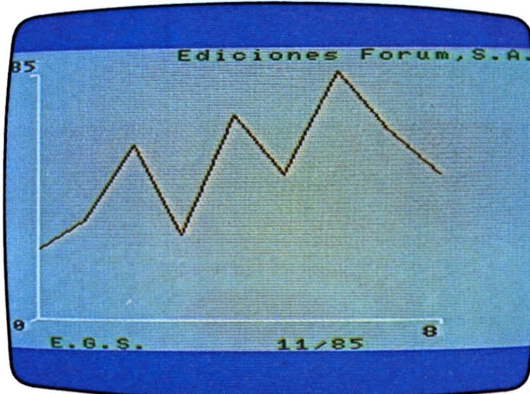
Ejemplo de programa: el número de datos que puede presentarse simultáneamente (o de columnas) está limitado por el espacio útil en la pantalla.



La versión MSX de este programa permite la presentación de gráficos en varios colores. Para la elección del color con que se desea trazar el gráfico (o el histograma) se ha previsto el menú secundario mostrado en esta foto.



Ejemplo de gráfico: después de introducir algunos datos, se ha pedido su gráfico. El lenguaje MSX permite presentar con facilidad caracteres en la pantalla de alta resolución, presentando cabeceras y valores indicativos.



Extensiones y modificaciones

Al ilustrar las distintas subrutinas ya se han sugerido algunas implantaciones de carácter secundario, por lo que en esta sección sólo se indicarán algunas de carácter sustancial. El procedimiento presentado permite la generación de gráficos en dos formas: gráfico continuo, también llamado gráfico X-Y, e histograma. El primero es útil para presentar fenómenos que pueden asimilarse a funciones. El segundo es más adecuado a los casos en que, conceptualmente, no puede existir una función o, en el mejor de los casos, en que el valor de la variable independiente (eje X) sólo asume algunos valores definidos y no admite intermedios.
Un ejemplo típico en el que debe utilizarse el segundo tipo de representación es la presentación de un balance (entradas y salidas).
Los datos están constituidos por los importes (de entrada o de salida) correspondientes al intervalo de tiempo elegido para el muestreo. Por ejemplo, definiendo un mes como período de observación, el gráfico estará constituido por 12 valores (para los 12 meses del año) y, como tales, o sea como puntos aislados, deberán ser presentados.
Si se unen entre sí los 12 puntos para obtener un gráfico X-Y, se cometerá un error, puesto que inducirá a creer que las observaciones se han realizado con una frecuencia mayor que la real, por ejemplo diaria.
Además de las dos formas de representación vistas e incluidas en el procedimiento, existen otras más adecuadas para mostrar algunos tipos particulares de fenómenos.
La más utilizada, no incluida en el procedimiento y que puede ser objeto de implantación, es el llamado «diagrama de tarta». Este tipo de gráfico es útil cuando se quiere mostrar la evolución de un conjunto que indique la influencia de distintos factores sobre el mismo fenómeno.
Por ejemplo, supongamos que se pretende analizar las componentes del costo de un producto.
De manera muy simplificada son:

■ Gastos para las materias primas	30%
■ Gastos de transformación	40%
■ Transportes	5%
■ Acopios	5%
■ Gastos para adecuación instalaciones	20%

Al lado de cada concepto aparece su incidencia porcentual sobre el costo del producto terminado. El gráfico necesario tiene la finalidad de presentar de manera sintética estos valores, poniendo de manifiesto su contribución sobre el total.
Ni la forma X-Y ni los histogramas se prestan a resolver este problema, porque los datos no poseen los necesarios atributos por estar constituidos por 5 valores, uno por concepto, en los cuales faltan los correspondientes valores de la variable independiente.
La única forma de presentarlo consiste en construir una figura que represente la totalidad (en este caso el 100%) poniendo de manifiesto en ella las distintas distribuciones. Esta figura puede ser cualquiera, incluso un rectángulo, pero generalmente se utiliza el círculo.
Suponiendo que se represente el costo de un producto con un rectángulo de base igual a 100, los conceptos unitarios estarán representados por otras tantas subdivisiones de base 30, 40, etc. Esta representación no permite una interpretación inmediata porque, si bien la altura del rectángulo puede representar otro parámetro, no se utiliza y esto produce incertidumbre en el observador.
Como se ha mencionado, la figura de base debe ser el círculo y los valores de cada variable se presentan con sectores de amplitud proporcional.
Por tanto, la implantación se reduce a presentar un círculo de radio cualquiera, presentando en el mismo tantos sectores como cuantas magnitudes en observación hay y con una apertura proporcional al valor correspondiente.
En el ejemplo anterior, los valores son porcentuales y, por tanto, su suma vale 100 y la apertura angular del sector que la representa viene dada por 3.6* valor en grados.
Tal y como se presenta, la implantación resuelve sólo los casos más sencillos. De hecho, la representación de un fenómeno con los diagramas «de tarta» se utiliza generalmente para poner de manifiesto valores relativos a comparaciones entre distintas situaciones y distribuciones.
Por ejemplo, en el caso de que se trate de representar el balance de una sociedad, los conceptos podrían ser los siguientes:

- Gastos para la producción
- Inversiones
- Investigación
- Pasivo

Representándolas en un histograma se tendría la presentación del peso de cada uno con respecto a los demás, pero no tan fácilmente lo que cada concepto pesa sobre el total. Para esta representación es más adecuado el diagrama de tarta, en el que cada sector tiene una amplitud proporcional a la cifra.
Utilizando esta forma se indica inmediatamente cuál de los conceptos tiene el importe más elevado señalando el sector que le corresponde o poniéndolo de manifiesto de manera diferente a los demás.




```
10 REM        ------------------
15 REM          GRAFICOS  C-64
25 REM        ------------------
26 POKE 52,140:POKE 56,140
30 MX=300:MY=190
40 MR=100:M3=20:DIMB$(MR),D$(M3),TP(M3),LC(M3),XC(M3),YC(M3),XD(M3),YD(M3)
50 DIMY(MR),X(MR),X1(MR),Y1(MR)
60 H1$="▀.I.♥.              DICIONES  ORUM. ♥.♠.":HO$=""
65 H2$="────────────────────────────────────────":HO$=""
70 FORI=1TO40:SP$=SP$+" ":NEXT
80 GOTO400
100 REM * PREPARA BIT-MAP *
101 POKE 53280,6
103 TQ=32768
105 POKE 56576,PEEK(56576)AND252
110 POKE 53272,8
115 POKE 53265,PEEK(53265)OR32
120 FOR K=TQ+24576 TO TQ+32575:POKE K,0
121 PK=K-21384-9200*(K>62343):POKE PK,0:NEXT K
125 FOR K=TQ+16384 TO TQ+17383:POKE K,230:NEXT K:REM 230=AZUL Y CELESTE
130 FOR K=0 TO 7:BI%(K)=2↑(7-K):NEXT K
135 RETURN
199 REM
200 REM * ENCIENDE PUNTO EN XS,YS *
201 REM
210 PU=TQ+24576+(40*(YSAND248)+(YSAND7)+(XSAND504))
211 PK=PU-21384-9200*(PU>62343)
214 ZS=PEEK(PK)ORBI%(XSAND7):POKE PU,ZS
215 POKE PK,ZS
220 RETURN
400 REM   ------
402 REM   -MAIN-
404 REM   ------
405 PRINTCHR$(14)
410 GOSUB1000
420 GOSUB2500
430 GOSUB4500
440 PRINTCHR$(14);:GOSUB2000:X=2:Y=14:GOSUB4160:PRINT"LECTA DE REGRESION "
450 X=25:GOSUB4160:PRINT"( / ) ?"
460 GETA$:IFA$="/"ORA$="N"THEN500
470 IFA$<>"S"ANDA$<>"♥"THEN460
490 GOSUB5000
500 GOSUB2000:X=11:Y=14:GOSUB4160:PRINT"          ( / / ) ?"
510 GETA$:IFA$="/"ORA$="N"THENPRINTHO$;CHR$(142);CHR$(154):END
520 IFA$<>"♥"ANDA$<>"S"THEN510
530 FORI=1TOMR:Y(I)=0:X(I)=0:X1(I)=0:Y1(I)=0:NEXTI:MM=SW(7)
540 IFKV=1THENKV=0:GOTO410
550 GOTO420
799 REM
800 REM * RESTABLECIMIENTO *
801 REM
810 POKE 56576,(PEEK(56576)AND252)OR3
815 POKE 53265,PEEK(53265)AND223
820 POKE 53272,21
825 PRINTHO$;CHR$(14):POKE53280,14:RETURN
1000 REM
1010 REM  LECTURA FICHERO
1020 REM
1030 GOSUB2000:X=1:Y=12:GOSUB4160:PRINT" 1  LOS DATOS ESTAN EN EL DISCO"
1040 Y=14:GOSUB4160:PRINT" 2  LOS DATOS SE INTRODUCIRAN DEL VIDEO"
1050 X=X+2:Y=17:GOSUB4160:PRINT"-UAL? "
1060 GETA$:IFA$<>"1"ANDA$<>"2"THEN1060
1070 PRINTA$;""
1080 IFA$="2"THENGOSUB3500:RETURN
1090 X=1:Y=21:GOSUB4160:INPUT"OMBRE DEL FICHERO  ";NF$
1095 IF LEN(NF$)=0 THEN 1000
1100 GOSUB2000:X=7:Y=12:GOSUB4160:PRINT"EL FICHERO HA SIDO GENERADO";
1105 Y=14:GOSUB4160:PRINT"POR LA          ( / / ) ?"
1110 GETA$:IFA$="♥"ORA$="S"THENGOSUB2200:I0=2:SW(1)=1:P7=2:GOTO1140
1120 IFA$<>"/"ANDA$<>"N"THEN1110
1130 GOSUB1500:SW(1)=0
1140 IFMM>MRTHENGOSUB4000
1145 GOSUB2000:X=7:Y=10:GOSUB4160:PRINT"INTRODUCIR EL DISCO DATOS":Y=Y+2:GOSUB41
60
1147 PRINT"     PULSAR UNA TECLA"
1150 GETA$:IFA$=""THEN1150
1155 LR=P7-1:FOR K=1 TO 20:LR=LR+LC(K):NEXT K
1160 OPEN15,8,15,"I/0"
1165 OPEN2,8,3,"0:"+NF$+",SEQ,R":INPUT#15,AD,X$,CD,DD
1167 IF AD=64 THEN CLOSE2:GOTO 1350
1170 IF AD=62 THEN CLOSE2:CLOSE15:GOTO 1000
1175 GOSUB 1300
1200 REM
1210 IF SW(1)=1 THEN INPUT#2,A$:MM=VAL(A$):IF MM>MR THEN GOSUB 4000
1215 IF SW(1)=1 THEN 1240
1220 IF I0=1 THEN 1240
1230 FOR I=1 TO I0-1:GOSUB 1400:NEXT:A$=""
1240 A=0:K1=1
1250 GOSUB1400:B$(K1)=A$
1260 IF SW(3)=3 AND K1<MM THEN FORI=1TOSW(4)-1:GOSUB1400:NEXTI:A$="":MM=MM-I+1
1270 IFSW(3)=4THENA=A+1:IFA=SW(4)-1ANDK1<MMTHENGOSUB1400:A$="":A=0:MM=MM-1
1280 K1=K1+1:IFK1<=MMTHEN1250
1290 CLOSE2:CLOSE15:SW(7)=MM:RETURN
1300 REM
1310 REM ESPERAR
1320 REM
1330 GOSUB2000:X=13:Y=15:GOSUB4160:PRINT"  ♥ ┐ ‾ _ ♠ _":RETURN
1350 REM **
1352 OPEN2,8,3,NF$:INPUT#15,AD,X$,CD,DD
1354 IF AD=62 THEN CLOSE2:CLOSE15:GOTO1000
1356 GOSUB 1300
1358 REM
1360 IF SW(1)=1 THEN CLOSE2:CLOSE15:GOTO1000
1362 RC=I0-1:IF RC<1 THEN RC=1
1364 A=0:K1=1
1370 GOSUB 1390:GOSUB1400:B$(K1)=A$:INPUT#15,AD,X$,CD,DD
1374 IFSW(3)=3ANDK1<MM THEN RC=RC+SW(4)-1
1376 IFSW(3)=4THENA=A+1:IFA=SW(4)-1ANDK1<MM THEN RC=RC+1:A=0
1380 K1=K1+1:RC=RC+1:IF RC<=MM THEN 1370
1385 CLOSE2:CLOSE15:SW(7)=MM:RETURN
1390 REM APUNTA AL REGISTRO#RC
1392 R1=RC:R2=0
1394 IF R1>256 THEN R2=INT(RC/256):R1=R1-256*R2
1396 PRINT#15,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
1398 RETURN
1400 REM LECTURA REGISTRO
1405 TT=1:GET#2,A$:IF A$=CHR$(13) THEN A$="":TT=0
1410 FOR R1=1 TO LR-TT
1412 GET#2,C$:A$=A$+C$
1414 NEXT R1
1416 RETURN
1500 REM INTRODUCCION FORMATO
1510 GOSUB2000:X=3:Y=10:GOSUB4160:PRINT"INTRODUCIR EL NUMERO "
1512 Y=Y+2:GOSUB4160:PRINT"DEL PRIMER REGISTRO UTIL"
1515 X=27:GOSUB4160:INPUTI0:IFI0<1THEN1510
1520 GOSUB2000:X=6:Y=9:GOSUB4160:PRINT"                    :"
1530 X=3:Y=18:GOSUB4160:INPUT"BYTE DE INICIO  ";P7:IFP7<1THEN1530
1540 FORI=1TOM3
1550 X=0:FORY=11TO23:GOSUB4160:PRINTLEFT$(SP$,39);:NEXTY
1560 X=13:Y=14:GOSUB4160:PRINT"CAMPO N.";I
1580 X=1:Y=19:GOSUB4160:INPUT"  TIPO (3=ALF. 1=NUM. -1=FIN)";TP(I)
1585 IFTP(I)=-1THEN1610
1587 IFTP(I)<>1ANDTP(I)<>3THEN1580
1590 X=9:Y=20:GOSUB4160:INPUT"LONGITUD";LC(I):IFLC(I)<1THEN1590
1600 NEXT
1610 M3=I-1
1620 GOSUB2000:X=0:Y=13:GOSUB4160:INPUT"UANTOS REGISTROS HAY EN EL FICHERO";M
M
1625 MM=MM+1-I0
1630 RETURN
2000 REM CABECERA
2010 PRINTHO$:X=2:Y=0:GOSUB4160:PRINTH1$
```

```
2015 X=2:Y=1:GOSUB4160:PRINTH2$
2020 X=11:Y=4:GOSUB4160:PRINT"▒┌─────────────────────┐"
2030 Y=5:GOSUB4160:PRINT"▒│ | _ ♠ _ ╲ ─ ┌ ♥ |"
2040 Y=6:GOSUB4160:PRINT"▒└─────────────────────┘▒"
2050 RETURN
2200 REM
2210 REM LECTURA FORMATO
2220 REM
2230 PRINTHO$:X=5:Y=7:GOSUB4160:PRINT"INTRODUCIR EL DISCO QUE CONTIENE"
2234 Y=9:GOSUB4160:PRINT"LAS DEFINICIONES DE LOS CAMPOS"
2240 GETA$:IFA$=""THEN2240
2250 GOSUB1300
2260 OPEN2,8,3,"0:DEF0,SEQ,R"
2270 FORI=1TOM3
2280 INPUT#2,R$
2290 XD(I)=VAL(MID$(R$,1,2))
2300 YD(I)=VAL(MID$(R$,3,2))
2310 D$(I)=MID$(R$,5,10)
2320 XC(I)=VAL(MID$(R$,15,2))
2330 YC(I)=VAL(MID$(R$,17,2))
2340 LC(I)=VAL(MID$(R$,19,2))
2350 TP(I)=VAL(MID$(R$,21,2))
2360 NEXTI:CLOSE2
2370 RETURN
2500 REM
2510 REM ELECCION DE LOS CAMPOS
2520 REM
2530 GOSUB5200
2540 IFKV=1THEN RETURN
2550 PRINTHO$:X=0:Y=0:GOSUB4160:PRINTH1$
2560 IFSW(1)=0THEN2610
2570 FORI=1TOM3:X=XD(I):Y=YD(I):GOSUB4160:PRINT"▒";D$(I):NEXT
2580 FORI=1TOM3:IF TP(I)=0THEN2600
2590 X=XC(I):Y=YC(I):GOSUB4160:PRINT"▒";STR$(I)
2600 NEXTI:GOTO2630
2610 X=1:FORY=1TOM3:GOSUB4160:PRINT"▒CAMPO N.";Y;TAB(15);"▒TIPO";TP(Y);
2615 PRINTTAB(25);"▒LONGITUD";LC(Y):NEXTY
2620 X=0:Y=22:GOSUB4160:PRINTLEFT$(SP$,39);
2630 X=0:Y=22:GOSUB4160:INPUT"▒NUMERO CAMPO ´▒▒´(▒/▒=PROGRES.)";X$:A=VAL(X$)
2640 IFX$="N"ORX$="/"THEN2670
2650 IFA<1ORA>M3THEN2620
2660 IFTP(A)=0ORTP(A)=3THEN2620
2670 X=0:Y=22:GOSUB4160:PRINTLEFT$(SP$,39);
2680 GOSUB4160:INPUT"NUMERO CAMPO ´▒I▒´  ";Y$:A=VAL(Y$):IFA<1ORA>M3THEN2670
2690 IFTP(A)=0ORTP(A)=3ORY$=X$THEN2670
2700 X=0:Y=22:GOSUB4160:PRINT"COORD.´♠´=CAMPO ";X$;"   COORD.´|´=CAMPO ";Y$;
2710 X=12:Y=23:GOSUB4160:PRINT"SEGURO (▒S/N▒)";
2720 GETA$:IFA$="/"ORA$="N"THEN2500
2730 IFA$<>"S"ANDA$<>"♥"THEN2720
2740 KX=VAL(X$):KY=VAL(Y$)
2750 FORJ=1TO2:IFJ=2THENA=KX:GOTO2770
2760 IFKX=0THEN2800
2765 A=KX
2770 P=P7:FORI=1TOA
2780 L=LC(I):P=P+L:NEXTI:P=P-L
2790 IFJ=1THENP1=P:L1=L
2795 IFJ<>1THENP2=P:L2=L
2800 NEXTJ
2810 FORI=1TOMM
2820 A2$=MID$(P$(I),P2,L2):Y(I)=VAL(A2$)
2830 IFKX=0THENX(I)=I:GOTO2850
2840 A1$=MID$(P$(I),P1,L1):X(I)=VAL(A1$)
2850 NEXTI
2860 IFMM>300THENGOSUB3000:REM AGREGACION
2870 RETURN
3000 REM
3010 REM  AGREGACION
3020 REM
3030 GOSUB2000
3040 X=1:Y=12:GOSUB4160:INPUT"NUMERO VALORES EJE ´X´(MAX 300)";N1
3043 IFN1>300ORN1<1THEN3000
3050 K=MM/N1
3060 I=0:FORZ=1TOMMSTEPK:I=I+1
3070 S1=0:S2=0:K1=0
3080 FORV=ZTOZ+K
3090 S1=S1+X(V):S2=S2+Y(V):K1=K1+1
3100 NEXTV
3110 X(I)=S1/K1:Y(I)=S2/K1
3120 NEXTZ
3130 FORJ=I+1TOMM:X(J)=0:Y(J)=0:NEXT
3140 MM=I:RETURN
3500 REM
3510 REM   LECTURA DATOS EN VIDEO
3520 REM
3530 KV=1:MM=0:GOSUB2000
3540 X=10:Y=10:GOSUB4160:PRINT"▒UANTOS DATOS ? (MAX 100)"
3550 X=13:Y=13:GOSUB4160:INPUT"▒";MM
3560 IFMM>100ORMM<2THEN3500
3570 X=4:Y=15:GOSUB4160:PRINT"1▒) X = VALOR PROGRESIVO"
3580 X=4:Y=17:GOSUB4160:PRINT"▒2▒) X = VALOR DADO DE TECLADO"
3590 X=11:Y=20:GOSUB4160:PRINT"▒UAL? ";
3600 GETA$:IFA$<>"1"ANDA$<>"2"THEN3600
3610 FORI=1TOMM:X=0:FORY=10TO21:GOSUB4160:PRINTLEFT$(SP$,39);:NEXTY
3620 X=15:Y=11:GOSUB4160:PRINT"DATO N.";RIGHT$(STR$(I),3)
3630 Y=12:GOSUB4160:PRINT"▒──────────▒"
3640 IFA$="1"THENX(I)=I-1:GOTO3660
3650 X=4:Y=16:GOSUB4160:INPUT"INTRODUCIR EL VALOR DE LA X";X(I):IFX(I)<0THEN3650
3660 X=4:Y=18:GOSUB4160:INPUT"INTRODUCIR EL VALOR DE LA Y";Y(I):IFY(I)<0THEN3660
3670 NEXTI:RETURN
4000 REM
4010 REM  SELECCION
4020 REM
4030 A=MM:A5=A/MR:A3=INT(A/(A-MR)):A2=INT(A6):IFA6<>INT(A5)THENA2=A2+1
4040 GOSUB2000:X=2:Y=9:GOSUB4160:PRINT"EL NUMERO DE REGISTROS ES SUPERIOR"
4042 Y=11:GOSUB4160:PRINT"AL MAXIMO ACEPTABLE (";MR;")"
4050 X=5:Y=15:GOSUB4160:PRINT"▒ELIJA EL CRITERIO DE SELECCION:"
4060 X=1:Y=17:GOSUB4160:PRINT"1) DESCARTAR LOS PRIMEROS";A-MR;"REGISTROS"
4070 Y=18:GOSUB4160:PRINT"2) DESCARTAR LOS ULTIMOS";A-MR;"REGISTROS"
4080 Y=19:GOSUB4160:PRINT"3) LEER UN REGISTRO CADA";A2
4090 A$="3":IFA3>1THENY=20:GOSUB4160:PRINT"4) DESCARTAR UN REGISTRO CADA";A3:A$=
"4"
4100 GETB$:IFB$<"1"ORB$>A$THEN4100
4110 IFB$="1"THENI0=I0+A-MR:MM=MR
4120 IFB$="2"THENMM=MR
4130 SW(3)=VAL(B$):IFSW(3)=3THENSW(4)=A2
4140 IFSW(3)=4THENSW(4)=A3
4150 RETURN
4160 REM POSICIONA EN X Y
4170 POKE211,X:POKE214,Y
4180 SYS 58732:RETURN
4500 REM
4510 REM   GRAFICOS
4520 REM
4530 GOSUB2000:X=11:Y=12:GOSUB4160:PRINT"▒▒ 1 ▒▒ HISTOGRAMA"
4540 Y=15:GOSUB4160:PRINT"▒▒ 2 ▒▒ GRAFICO":X=14:Y=20:GOSUB4160:PRINT"▒UAL?"
4550 GETA$:IFA$<>"1"ANDA$<>"2"THEN4550
4630 IFA$="2"THENSW(2)=1:GOTO4650
4640 SW(2)=0:IFMM>14THENN1=14:GOSUB1300:GOSUB3050
4650 XM=X(1):YM=Y(1)
4660 FORI=2TOMM:IFX(I)>XMTHENXM=X(I)
4670 IFY(I)>YMTHENYM=Y(I)
4680 NEXTI
4684 IF YM=0 THEN YM=1E-5
4685 IF XM=0 THEN XM=1E-5
4690 FORI=1TOMM
4700 X(I)=24+X(I)*280/XM
4710 Y(I)=15+Y(I)*155/YM:Y(I)=185-Y(I)
4720 NEXT
4730 GOSUB100
4750 X1=23:X2=23:Y1=15:Y2=171:GOSUB5600
4770 X1=23:X2=305:Y1=171:Y2=171:GOSUB5600
4780 X1=19:X2=22:Y1=15:Y2=15:GOSUB5600
4790 X1=19:X2=22:Y1=173:Y2=171:GOSUB5600
4840 IFSW(2)=0THENGOSUB4920:GOTO4970
4850 X1=305:X2=305:Y1=172:Y2=174:GOSUB5600
4880 FORI=1TOMM-1
4890 X1=X(I):X2=X(I+1):Y1=Y(I):Y2=Y(I+1):GOSUB5600
4900 NEXT:GOTO4970
4920 REM HISTOGRAMA
4930 L=10*14/MM
4940 A=24:FORI=1TOMM
4950 X1=A:X2=A:Y1=170:Y2=Y(I):GOSUB5600
4952 X1=A:X2=A+L:Y1=Y(I):Y2=Y(I):GOSUB5600
4955 X1=A+L:X2=A+L:Y1=Y(I):Y2=170:GOSUB5600
4960 A=L+L+A:NEXT:RETURN
4970 GETA$:IFA$=""THEN4970
4980 GOSUB800:IFSW(2)=0THENGOSUB5300
4990 RETURN
```

```
5000 REM
5010 REM    RECTA DE REGRESION
5020 REM
5030 N=MM:A1=0:A2=0:A3=0:A4=0
5040 FORI=1TON
5050 A1=A1+X1(I):A2=A2+Y1(I):A4=A4+X1(I)*Y1(I):A3=A3+X1(I)↑2
5060 NEXTI
5070 D=N*A3-A1↑2:IFD=0THEN5152
5075 B=(A2*A3-A1*A4)/D
5080 A=(N*A4-A1*A2)/D
5085 IF INT(A+.49)=0THEN5152
5090 X0=INT(-B/A+.49):Y0=INT(B+.49)
5100 GOSUB2000:X=1:Y=10:GOSUB4160:PRINT"LA ECUACION DE LA RECTA DE REGRESION";
5105 Y=12:GOSUB4160:PRINT"ES: ■ | =";INT(A+.49);" * ♦ ";
5110 IF INT(B)=0THEN5120
5112 IF INT(B)>0THENPRINT"+";
5115 PRINTINT(B+.49);
5120 X=4:Y=16:GOSUB4150:PRINT"■LA RECTA CORTA:"
5130 X=4:Y=19:GOSUB4160:PRINT"■EL EJE X EN EL PUNTO X=■";X0
5140 X=4:Y=21:GOSUB4160:PRINT"■EL EJE Y■EN EL PUNTO Y=■";Y0
5150 X=11:Y=23:GOSUB4160:PRINT"■PULSAR UNA TECLA";:GOTO5160
5152 GOSUB2000:X=8:Y=12:GOSUB4160:PRINT"LA ECUACION DE LA RECTA"
5153 Y=14:GOSUB4160:PRINT"NO ES REPRESENTABLE"
5155 X=11:Y=20:GOSUB4160:PRINT"■PULSAR UNA TECLA"
5160 GETA$:IFA$=""THEN5160
5170 RETURN
5200 REM  SALVA LOS VALORES ORIGINALES
5230 FORI=1TOMM:X1(I)=X(I):Y1(I)=Y(I)
5240 NEXT:RETURN
5300 REM IMPRESION HISTOGRAMA
5310 GOSUB2000:X=1:Y=12:GOSUB4160:PRINT"DESEA LA IMPRESION DEL HISTOGRAMA?"
5320 X=15:Y=17:GOSUB4160:PRINT"(S/N)"
5330 GETA$:IFA$="N"ORA$="/"THENRETURN
5340 IFA$<>"S"ANDA$<>"●"THEN5330
5350 X=0:Y=18:GOSUB4160:PRINT"■POSICIONAR EL PAPEL Y PULSAR UNA TECLA"
5360 GETA$:IFA$=""THEN5360
5370 GOSUB2000:X=4:Y=10:GOSUB4160:PRINT"INTRODUCIR LA CABECERA DE IMPRESION"
5380 X=0:Y=13:GOSUB4160:INPUTA$
5385 OPEN4,4:CMD4
5390 PRINTCHR$(14);CHR$(17);H1$;CHR$(15):PRINT:PRINT:PRINTTAB((80-LEN(A$))/2);A$
5395 PRINTCHR$(145):PRINT:PRINT"|":PRINT"|"
5397 S$="":FORI=1TO80:S$=S$+"▒":NEXT
5400 FORI=1TOMM
5405 IFY(I)>170THENY(I)=170
5410 PRINT"|";LEFT$(S$,(170-Y(I))/3)
5420 PRINT"|";LEFT$(S$,(170-Y(I))/3)
5430 PRINT"|":PRINT"|":NEXT
5440 CMD4:PRINTTAB(52);"MAX=";YM
5450 PRINT#4:CLOSE4:RETURN
5500 REM TRAZA LINEA
5510 DX=INT(ABS(X1-X2)):DY=INT(ABS(Y1-Y2))
5620 IFDX=0ANDDY=0THENXS=X1:YS=Y1:GOSUB200:RETURN
5630 IFDX>DYTHEN5680
5640 FORYS=Y1TOY2STEPSGN(Y2-Y1)
5650 XS=(X2-X1)/(Y2-Y1)*(YS-Y1)+X1
5660 GOSUB200:NEXTYS
5670 RETURN
5680 FORXS=X1TOX2STEPSGN(X2-X1)
5690 YS=(Y2-Y1)/(X2-X1)*(XS-X1)+Y1
5700 GOSUB200:NEXTXS
5710 RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - COMMODORE 64

MR	= máx. número de registros aceptables por el fichero datos
M3	= máx. número de campos para cada registro
TP(*)	= tipo de los campos
LC(*)	= longitud de los campos
XC(*) **YC(*)**	= coordenadas de los campos
XD(*) **YD(*)**	= coordenadas de las descripciones
D$(*)	= descripciones
H1$(*)	= cabecera
Y(*) **X(*)**	= abscisas y ordenadas modificables para adecuarse al sistema de referencia
Y1(*) **X1(*)**	= abscisas y ordenadas originales no modificables
MX	= número de valores (abscisas y ordenadas) modificables durante la ejecución del programa
SW(7)	= valor original de MX (no modificable)
KV	= flag de datos introducidos por teclado/datos procedentes de cinta o de disco
SW(1)	= flag fichero de la Base de Datos/otro fichero
I0	= número del primer registro útil
P7	= primer byte útil del registro
NF$	= nombre del fichero
SW(3) **SW(4)**	= valores para la lectura del fichero seleccionado
SW(2)	= flag gráfico/histograma
YM	= máximo valor de la abscisa
XM	= máximo valor de la ordenada



```
10   REM      -----------
20   REM      GRAFICOS
25   REM      VERSION DOS
30   REM      -----------
40   D1$ =  CHR$ (4):
     BP$ =  CHR$ (7):
     B$ = "                "
45   DR$ = "1"
50   BA$ = B$ + B$ + B$ + B$:
     B$ = ""
55   B$ = "*******************":
     AS$ = B$ + B$ + B$ + B$:
     B$ = ""
80   MR = 130:
     M3 = 20:
     DIM B$(MR),D$(M3),TP(M3),LC(M3),X
     C(M3),YC(M3),XD(M3),YD(M3),Y(MR),
     X(MR),X1(MR),Y1(MR)
90   H1$ = "E.G.S.                EDICION
     ES FORUM,S.A."
200  REM   ----
210  REM   MAIN
220  REM   ----
230  GOSUB 1000
240  GOSUB 2500
250  GOSUB 4500
260  GOSUB 2000:
     X = 1:
     Y = 14:
     GOSUB 4160:
     PRINT "QUIERE VER LA RECTA DE REG
     RESION ?"
270  X = 14:
     Y = 16:
     GOSUB 4160:
     PRINT "(S/N)";
280  GET A$:
     IF A$ = "N" THEN 310
290  IF A$ <  > "S" THEN 280
300  GOSUB 5000
310  GOSUB 2000:
     X = 9:
     Y = 14:
     GOSUB 4160:
     PRINT "OTRO GRAFICO ? (S/N)";
320  GET A$:
     IF A$ = "N" THEN  GOSUB 2000:
     PRINT BP$;:
     X = 13:
     Y = 13:
     GOSUB 4160:
     PRINT "FIN PROGRAMA":
     END
330  IF A$ <  > "S" THEN 320
340  FOR I = 1 TO MR:
       Y(I) = 0:
       X(I) = 0:
       X1(I) = 0:
       Y1(I) = 0:
     NEXT :
     MX = SW(7)
350  IF KV = 1 THEN KV = 0:
     GOTO 230
360  GOTO 240
1000 REM  -------------------
1030 GOSUB 2000:
     X = 4:
     Y = 12:
     GOSUB 4160:
     PRINT "1) LOS DATOS ESTAN EN EL D
     ISCO"
1040 Y = 14:
     GOSUB 4160:
     PRINT "2) LOS DATOS SE ENTRARAN P
     OR VIDEO"
1050 X = X + 3:
     Y = 17:
     GOSUB 4160:
     PRINT "CUAL ?";
1060 GET A$:
     IF A$ <  > "1" AND A$ <  > "2"
      THEN 1060
1070 X = X + 7:
     GOSUB 4160:
     PRINT A$
1080 IF A$ = "2" THEN  GOSUB 3500:
     RETURN
1090 X = 1:
     Y = 21:
     GOSUB 4160:
     PRINT "* NOMBRE DEL FICHERO *";:
     INPUT FF$
1100 GOSUB 2000:
     X = 7:
     Y = 12:
     GOSUB 4160:
     PRINT "EL FICHERO LO HA GENERADO"
     ; CHR$ (10);"                LA <B
     ASE DE DATOS>? (S/N)";
1110 GET A$:
     IF A$ = "S" THEN  GOSUB 2200:
     IO = 2:
     SW(1) = 1:
     P7 = 2:
     LR = 128:
     GOTO 1140
1120 IF A$ <  > "N" THEN 1110
1130 GOSUB 1500:
     SW(1) = 0
1140 IF MX > MR THEN  GOSUB 4000
1150 REM  --- LEE DISCO ---
1160 GOSUB 2000:
     X = 9:
     Y = 12:
     GOSUB 4160:
     PRINT "INTRODUCIR DISCO DATOS"
1170 X = 9:
     Y = 20:
     GOSUB 4160:
     PRINT "DESPUES PULSAR UNA TECLA";

1180 GET A$
1190 GOSUB 1300
1200 NF$ = FF$:
     GOSUB 6000
1205 IF SW(1) = 0 THEN 1215
1210 NR = 1:
     GOSUB 6200:
     MX =  VAL (AA$):
     IF MX > MR THEN  GOSUB 4000
1215 A = 0:
     K1 = 1:
     NR = IO
1220 GOSUB 6200
1230 B$(K1) = AA$
1240 IF SW(3) = 3 AND K1 < MX THEN NR
      = NR + SW(4) - 1
1250 IF SW(3) = 4 THEN A = A + 1:
     IF A = SW(4) - 1 AND K1 < MX
      THEN NR = NR + 1:
     A = 0
1280 K1 = K1 + 1:
     NR = NR + 1:
     IF NR <  = MX THEN 1220
1290 GOSUB 6600:
     SW(7) = MX:
     RETURN
1300 REM  --- ESPERAR ---
1310 GOSUB 2000:
     X = 14:
     Y = 15:
     GOSUB 4160:
     PRINT "E S P E R A R":
     RETURN
1500 REM  --- INTROD.FORM.---
1510 GOSUB 2000:
     X = 0:
     Y = 10:
     GOSUB 4160:
     PRINT "INTRODUCIR EL NUMERO DEL P
     RIMER REG.UTIL"
1520 X = 14:
     Y = 14:
     GOSUB 4160:
     INPUT IO:
     IF IO < 1 THEN 1510
1525 GOSUB 2000:
     X = 6:
     Y = 9:
     GOSUB 4160:
     PRINT "INTRODUCIR EL FORMATO REGI
     STRO."
1530 X = 3:
     Y = 18:
     GOSUB 4160:
     PRINT "BYTE PRINCIPIO";:
     INPUT P7:
     IF P7 < 1 THEN 1530
1540 FOR I = 1 TO M3:
       X = 0:
       FOR Y = 11 TO 23:
         GOSUB 4160:
         PRINT  SPC( 39);:
       NEXT Y
1560   X = 13:
       Y = 14:
       GOSUB 4160:
       PRINT "CAMPO N. ";I
1570   Y = 15:
       GOSUB 4160:
       PRINT "-----------"
1580   X = 1:
       Y = 18:
       GOSUB 4160:
       PRINT "TIPO (3=ALF. 1=NUM. -1=
       FIN)";:
       INPUT TP(I):
       IF TP(I) =  - 1 THEN 1610
1585   IF TP(I) <  > 1 AND TP(I)
        <  > 3 THEN 1580
1590   X = 9:
       Y = 20:
       GOSUB 4160:
       PRINT "LONGITUD";:
       INPUT LC(I):
       IF LC(I) < 1 THEN 1590
1600 NEXT
1610 M3 = I - 1
1615 LR = 0:
     FOR I = 1 TO M3:
       LR = LR + LC(I):
     NEXT :
     LR = LR + P7 - 1
```

```
1620  GOSUB 2000:
      X = 0:
      Y = 13:
      GOSUB 4160:
      PRINT "CUANTOS REGISTROS CONTIENE
       EL FICHERO ";:
      INPUT MX:
      MX = MX + 1 - IO
1630  RETURN
2000  REM  --- INT.---
2010  TEXT :
      HOME :
      X = 0:
      Y = 0:
      GOSUB 4160:
      PRINT H1$
2020  X = 10:
      Y = 4:
      GOSUB 4160:
      PRINT "*******************"
2030  Y = 5:
      GOSUB 4160:
      PRINT "* G R A F I C O S *"
2040  Y = 6:
      GOSUB 4160:
      PRINT "*******************"
2050  RETURN
2200  REM  -- LET FORM.---
2230  HOME :
      X = 8:
      Y = 10:
      GOSUB 4160:
      PRINT "INTRODUCIR EL DISCO DATOS"
      :
      X = 9:
      Y = 20:
      GOSUB 4160:
      PRINT "DESPUES PULSAR UNA TECLA"
2240  GET A$
2250  GOSUB 1300:
      NF$ = "DEFC":
      LR = 26:
      GOSUB 6000
2260  FOR I = 1 TO M3:
        NR = I:
        GOSUB 6200
2270    XD(I) =  VAL ( MID$ (AA$,1,2))
        :
        YD(I) =  VAL ( MID$ (AA$,3,2))

2280    D$(I) =  MID$ (AA$,5,10)
2290    XC(I) =  VAL ( MID$ (AA$,15,2)
        ):
        YC(I) =  VAL ( MID$ (AA$,17,2)
        )
2300    LC(I) =  VAL ( MID$ (AA$,19,2)
        )
2310    TP(I) =  VAL ( MID$ (AA$,21,2)
        )
2320  NEXT I:
      GOSUB 6600
2370  RETURN
2500  REM  --- ELIGE CAMPO ---
2530  GOSUB 5200
2540  IF KV = 1 THEN  RETURN
2550  HOME :
      X = 0:
      Y = 0:
      GOSUB 4160:
      PRINT H1$
2560  IF SW(1) = 0 THEN 2610
2570  FOR I = 1 TO M3:
        IF D$(I) =  LEFT$ (BA$,10)
         OR D$(I) = "" THEN 2577
2575    X = XD(I):
        Y = YD(I):
        GOSUB 4160:
        PRINT D$(I)
2577  NEXT I
2580  FOR I = 1 TO M3:
        IF TP(I) = 0 THEN 2600
2590    X = XC(I):
        Y = YC(I):
        GOSUB 4160:
        PRINT " ";I
2600  NEXT I:
      GOTO 2630
2610  X = 1:
      FOR Y = 1 TO M3:
        GOSUB 4160:
        PRINT "CAMPO N. ";Y; TAB( 16);
        "TIPO ";TP(Y); TAB( 26);"LONGI
        TUD";LC(Y):
      NEXT Y
2620  X = 0:
      Y = 22:
      GOSUB 4160:
      PRINT  SPC( 39);
2630  X = 0:
      Y = 22:
      GOSUB 4160:
      PRINT "NUMERO CAMPO 'X'(N=PROGRES
      .) ";:
      INPUT X$:
      A =  VAL (X$)
2640  IF X$ = "N" THEN 2670
2650  IF A < 1 OR A > M3 THEN 2620
2660  IF TP(A) = 0 OR TP(A) = 3 THEN 26
      20
2670  X = 0:
      Y = 22:
      GOSUB 4160:
      PRINT  SPC( 39);
2680  X = 0:
      Y = 22:
      GOSUB 4160:
      PRINT "NUMERO CAMPO 'Y'  ";:
      INPUT Y$:
      A =  VAL (Y$):
      IF A < 1 OR A > M3 THEN 2670
2690  IF TP(A) = 0 OR TP(A) = 3 OR Y$
       = X$ THEN 2670
2700  X = 0:
      Y = 22:
      GOSUB 4160:
      PRINT "COORD.'X'=CAMPO ";X$;"   C
      OORD.'Y'=CAMPO ";Y$
2710  X = 12:
      Y = 23:
      GOSUB 4160:
      PRINT "SEGURO? (S/N)";
2720  GET A$:
      IF A$ = "N" THEN 2500
2730  IF A$ <  > "S" THEN 2720
2740  KX =  VAL (X$):
      KY =  VAL (Y$)
2750  FOR J = 1 TO 2:
        IF J = 2 THEN A = KY:
        GOTO 2770
2760    IF KX = 0 THEN 2800
2765    A = KX
2770    P = P7:
        FOR I = 1 TO A
2780      L = LC(I):
          P = P + L:
        NEXT I:
        P = P - L
2790    IF J = 1 THEN P1 = P:
        L1 = L:
        GOTO 2800
2795    P2 = P:
        L2 = L
2800  NEXT J
2810  FOR I = 1 TO MX
2820    A2$ =  MID$ (B$(I),P2,L2):
        Y(I) =  VAL (A2$)
2830    IF KX = 0 THEN X(I) = I:
        GOTO 2850
2840    A1$ =  MID$ (B$(I),P1,L1):
        X(I) =  VAL (A1$)
2850  NEXT I
2860  IF MX > 200 THEN  GOSUB 3000:
      REM  --- AGREGACION ---

2870  RETURN
3000  REM  --- COMPRESOR ---
3030  GOSUB 2000:
      PRINT BF$
3040  X = 1:
      Y = 12:
      GOSUB 4160:
      PRINT "NUMERO VALORES EJES 'X'(MA
      X 200)";:
      INPUT N1:
      IF N1 > 200 OR N1 < 1 THEN 3000
3050  K = MX / N1
3060  I = 0:
      FOR Z = 1 TO MX STEP K:
        I = I + 1
3070    S1 = 0:
        S2 = 0:
        K1 = 0
3080    FOR V = Z TO Z + K
3090      S1 = S1 + X(V):
          S2 = S2 + Y(V):
          K1 = K1 + 1
3100    NEXT V
3110    X(I) = S1 / K1:
        Y(I) = S2 / K1
3120  NEXT Z
3130  FOR J = I + 1 TO MX:
        X(J) = 0:
        Y(J) = 0:
      NEXT
3140  MX = I:
      RETURN
3500  REM  --- LECT.DATOS VIDEO ---
3530  KV = 1:
      MX = 0:
      GOSUB 2000
3540  X = 1:
      Y = 10:
      GOSUB 4160:
      PRINT "CUANTOS DATOS DESEA INTROD
      .? (MAX 130)"
3550  X = 13:
      Y = 13:
      GOSUB 4160:
      INPUT MX:
      IF MX > MR OR MX < 2 THEN 3500
3570  X = 4:
      Y = 15:
      GOSUB 4160:
      PRINT "1) X = VALOR PROGRESIVO"
3580  Y = 17:
      GOSUB 4160:
      PRINT "2) X = VALOR DATO DE TECLA
      DO"
3590  X = 11:
      Y = 20:
      GOSUB 4160:
      PRINT "CUAL ?";
3600  GET A$:
      IF A$ <  > "1" AND A$ <  > "2"
       THEN 3600
3610  FOR I = 1 TO MX:
        X = 0:
        FOR Y = 10 TO 21:
          GOSUB 4160:
          PRINT  SPC( 39);:
        NEXT Y
3620    X = 15:
        Y = 11:
        GOSUB 4160:
        PRINT "DATO N."; RIGHT$ (" "
         +  STR$ (I),3)
3630    Y = 12:
```

```
         GOSUB 4160:
         PRINT "----------"
3640     IF A$ = "1" THEN X(I) = I
          - 1:
         GOTO 3660
3650     X = 4:
         Y = 16:
         GOSUB 4160:
         PRINT "INTRODUCIR EL VALOR DE
         LA X"::
         INPUT X(I):
         IF X(I) < 0 THEN 3650
3660     X = 4:
         Y = 18:
         GOSUB 4160:
         PRINT "INTRODUCIR EL VALOR DE
         LA Y"::
         INPUT Y(I):
         IF Y(I) < 0 THEN 3660
3670  NEXT I:
      RETURN
4000  REM  --- SELECCION ---
4030  A = MX:
      A6 = A / MR:
      A3 =  INT (A / (A - MR)):
      A2 =  INT (A6):
      IF A6 <  >  INT (A6) THEN A2
       = A2 + 1
4040  GOSUB 2000:
      X = 3:
      Y = 9:
      GOSUB 4160:
      PRINT "EL NUMERO DE REGISTROS ES
      SUPERIOR "; CHR$ (10);"       AL M
      AXIMO ACEPTABLE (";MR;")"
4050  X = 5:
      Y = 15:
      GOSUB 4160:
      PRINT "ELIJA EL CRITERIO DE SELEC
      CION."
4060  X = 1:
      Y = 17:
      GOSUB 4160:
      PRINT "1) DESC.LOS PRIMEROS ";A
       - MR;" REGISTROS UTILES"
4070  Y = 18:
      GOSUB 4160:
      PRINT "2) DESC.LOS ULTIMOS ";A
       - MR;" REGISTROS"
4080  Y = 19:
      GOSUB 4160:
      PRINT "3) LEER UN REGISTRO CADA "
      ;A2
4090  A$ = "3":
      IF A3 > 1 THEN Y = 20:
      GOSUB 4160:
      PRINT "4) DESC.UN REGISTRO CADA "
      ;A3:
      A$ = "4"
4100  GET B$:
      IF B$ < "1" OR B$ > A$ THEN 4100
4110  IF B$ = "1" THEN IO = IO + A
       - MR:
      MX = MR
4120  IF B$ = "2" THEN MX = MR
4130  SW(3) =  VAL (B$):
      IF SW(3) = 3 THEN SW(4) = A2
4140  IF SW(3) = 4 THEN SW(4) = A3
4150  RETURN
4160  REM  --- POSICIONA CURSOR ---
4170  HTAB X + 1:
      VTAB Y + 1:
      RETURN
4500  REM  GRAFICI
4530  GOSUB 2000:
      X = 11:
      Y = 12:
      GOSUB 4160:
      PRINT "1) HISTOGRAMA"
4540  Y = 15:
      GOSUB 4160:
      PRINT "2) GRAFICO":
      X = 14:
      Y = 20:
      GOSUB 4160:
      PRINT "CUAL ?";
4550  GET A$:
      IF A$ <  > "1" AND A$ <  > "2"
       THEN 4550
4630  IF A$ = "2" THEN SW(2) = 1:
      GOTO 4650
4640  SW(2) = 0:
      IF MX > 14 THEN N1 = 14:
      GOSUB 1300:
      GOSUB 3050
4650  XM = X(1):
      YM = Y(1)
4660  FOR I = 2 TO MX:
        IF X(I) > XM THEN XM = X(I)
4670    IF Y(I) > YM THEN YM = Y(I)
4680  NEXT I
4690  FOR I = 1 TO MX:
        X(I) = 24 + X(I) * 180 / XM
4700    Y(I) = 15 + Y(I) * 155 / YM:
        Y(I) = 185 - Y(I)
4710  NEXT I
4720  HGR2 :
      HCOLOR= 3
4750  A = 171:
      HPLOT 23,15 TO 23,A
4760  D = 0:
      IF SW(2) = 0 THEN D = 10
4770  HPLOT 23,A TO 205 + D,A
4780  HPLOT 19,15 TO 22,15
4790  HPLOT 19,A + 2 TO 22,A
4840  IF SW(2) = 0 THEN  GOSUB 4920:
      GOTO 4970
4850  HPLOT 205,A + 1 TO 205,A + 3
4880  FOR I = 1 TO MX - 1
4890    HPLOT X(I),Y(I) TO X(I + 1),Y(
        I + 1)
4900  NEXT
4910  GOTO 4970
4920  REM  --- HISTOGRAMA ---
4930  L = 7 * 14 / MX
4940  A = 24:
      FOR I = 1 TO MX
4945    IF Y(I) < 0 THEN Y(I) = 0
4947    IF Y(I) > 191 THEN Y(I)
         = 191
4950    HPLOT A,170 TO A,Y(I) TO A
         + L,Y(I) TO A + L,170
4960    A = A + L + L:
      NEXT :
      RETURN
4970  GET A$
4980  TEXT :
      IF SW(2) = 0 THEN  GOSUB 5300
4990  RETURN
5000  REM  --- RECTA REGRESION ---
5030  N = MX:
        A1 = 0:
        A2 = 0:
        A3 = 0:
        A4 = 0
5040  FOR I = 1 TO N:
        A1 = A1 + X1(I):
        A2 = A2 + Y1(I):
        A4 = A4 + X1(I) * Y1(I):
        A3 = A3 + X1(I) ^ 2
5060  NEXT I
5070  D = N * A3 - A1 ^ 2:
      IF D = 0 THEN 5250
5075  B = (A2 * A3 - A1 * A4) / D
5080  A = (N * A4 - A1 * A2) / D:
      IF  INT (A + .49) = 0 THEN 5250
5090  X0 =  - B / A:
      Y0 = B
5100  GOSUB 2000:
      X = 1:
      Y = 10:
      GOSUB 4160:
      PRINT "LA ECUACION DE LA RECTA DE
       REGRESION "; CHR$ (10);"  ES.  Y
      = "; INT (A + 0.49);"* X ";
5110  IF B = 0 THEN 5120
5112  IF B > 0 THEN  PRINT "+ ";
5115  PRINT  INT (B + 0.49);
5120  X = 4:
      Y = 16:
      GOSUB 4160:
      PRINT "LA RECTA CORTA. ";
5130  X = 4:
      Y = 19:
      GOSUB 4160:
      PRINT "EL EJE X EN EL PUNTO X=";
       INT (X0 + 0.49);
5140  Y = 21:
      GOSUB 4160:
      PRINT "EL EJE Y EN EL PUNTO Y=";
       INT (Y0 + 0.49)
5150  Y = 23:
      X = 12:
      GOSUB 4160:
      PRINT "PULSAR UNA TECLA";
5160  GET A$
5170  RETURN
5200  REM  --- SALV.VALOR ORIGINAL ---

5210  FOR I = 1 TO MX:
        X1(I) = X(I):
        Y1(I) = Y(I)
5220  NEXT I:
      RETURN
5250  HOME :
      HTAB 5:
      VTAB 13:
      PRINT "ECUACION NO REPRESENTABLE"
      :
      HTAB 10:
      VTAB 20:
      PRINT "PULSAR UNA TECLA "::
      GET A$:
      RETURN
5300  REM  --- IMPRESION HISTOGRAMA ---

5310  GOSUB 2000:
      X = 1:
      Y = 12:
      GOSUB 4160:
      PRINT "DESEA LA IMPRESION DEL HIS
      TOGRAMA ?"
5320  X = 15:
      Y = 15:
      GOSUB 4160:
      PRINT "(S/N)";
5330  GET A$:
      IF A$ = "N" THEN  RETURN
5340  IF A$ <  > "S" THEN 5330
5350  X = 0:
      Y = 18:
      GOSUB 4160:
      PRINT "POSICIONAR EL PAPEL Y PULS
      AR UNA TECLA";BP$;
5360  GET A$
5370  GOSUB 2000:
      X = 4:
      Y = 10:
      GOSUB 4160:
      PRINT "INTRODUCIR LA CABECERA DE
      IMPRESION"
5380  X = 0:
      Y = 13:
      GOSUB 4160:
      INPUT A$
5390  PRINT D1$:
      PRINT D1$;"PR#1":
```

```
PRINT D1$:
PRINT  CHR$ (9);"80N":
PRINT  CHR$ (14);H1$; CHR$ (18):
PRINT :
PRINT :
PRINT  TAB( (80 -  LEN (A$))
 / 2);A$:
        PRINT :
        PRINT :
        PRINT "I":
        PRINT "I"
5400  FOR I = 1 TO MX
5405     IF Y(I) > 170 THEN Y(I)
          = 170
5410     PRINT "I"; MID$ (AS$,1,(170
          - Y(I)) / 3)
5420     PRINT "I"; MID$ (AS$,1,(170
          - Y(I)) / 3)
5430     PRINT "I":
         PRINT "I":
      NEXT
5440  PRINT  TAB( 52);"MAX=";YM
5450  PRINT D1$:
      PRINT D1$;"PR#0":
      PRINT D1$:
      RETURN
6000  REM  *OPEN*
6030  PRINT D1$
6040  PRINT D1$;"OPEN";NF$ + ",D"
       + DR$;",L";LR + 2
6050  PRINT D1$
6060  RETURN
6200  REM  *GET*
6230  PRINT D1$
6240  PRINT D1$;"READ";NF$;",R";NR
6250  INPUT AA$
6260  PRINT D1$
6270  RETURN
6400  REM  *PUT*
6430  PRINT D1$
6440  PRINT D1$;"WRITE";NF$;",R";NR
6450  PRINT  CHR$ (34) + AA$
6460  PRINT D1$
6470  RETURN
6600  REM  *CLOSE*
6630  PRINT D1$
6640  PRINT D1$;"CLOSE";NF$
6650  PRINT D1$
6660  RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - APPLE II

MR	= máx. número de registros aceptables por el fichero datos
M3	= máx. número de campos para cada registro
TP(*)	= tipo de los campos
LC(*)	= longitud de los campos
XC(*), **YC(*)**	= coordenadas de los campos
XD(*), **YD(*)**	= coordenadas de las descripciones
D$(*)	= descripciones
H1$(*)	= cabecera
Y(*), **X(*)**	= abscisas y ordenadas modificables para adecuarse al sistema de referencia
Y1(*), **X1(*)**	= abscisas y ordenadas originales no modificables
MX	= número de valores (abscisas y ordenadas) modificables durante la ejecución del programa
SW(7)	= valor original de MX (no modificable)
KV	= flag de datos introducidos por teclado/datos procedentes de cinta o de disco
SW(1)	= flag de fichero de la Base de Datos/otro fichero
I0	= número del primer registro útil
P7	= primer byte útil del registro
NF$	= nombre del fichero
SW(3), **SW(4)**	= valores para la lectura del fichero seleccionado
SW(2)	= flag gráfico/histograma
YM	= máximo valor abscisa
XM	= máximo valor ordenada

RELACION DE LAS PRINCIPALES VARIABLES - PHILIPS VG 8010

MR	= máx. número de registros aceptables por el fichero datos
M3	= máx. número de campos para cada registro
TP(*)	= tipo de los campos
LC(*)	= longitud de los campos
XC(*), **YC(*)**	= coordenadas de los campos
XD(*), **YD(*)**	= coordenadas de las descripciones
D$(*)	= descripciones
H1$(*)	= cabecera
Y(*), **X(*)**	= abscisas y ordenadas modificables para adecuarse al sistema de referencia
Y1(*), **X1(*)**	= abscisas y ordenadas originales no modificables
MX	= número de valores (abscisas y ordenadas) modificables durante la ejecución del programa
SW(7)	= valor original de MX (no modificable)
KV	= flag de datos introducidos por teclado/datos procedentes de cinta o de disco
SW(1)	= flag fichero de la Base de Datos/otro fichero
I0	= número del primer registro útil
P7	= primer byte útil del registro
NF$	= nombre del fichero
SW(3), **SW(4)**	= valores para la lectura fichero seleccionado
SW(2)	= flag gráfico/histograma
YM	= máximo valor abscisa
XM	= máximo valor ordenada

Gráficos de ordenador/Versión Philips VG 8010

```
10  REM --------
20  REM GRAFICOS
25  REM Version MSX
30  REM -----------
50  MAXFILES=2
60  SCREEN0:COLOR15,4:CLEAR 8000:DEFINTI,J
70  KEYOFF:WIDTH40
80  MR=130:M3=20:DIMB$(MR),D$(M3),TP(M3),LC(M3),XC(M3),YC(M3),XD(M3),YD(M3),Y(MR),X(MR),X1(MR),
Y1(MR)
90  H1$="E.G.S.          Ediciones Forum, S.A."
100 BP$="O6L32CDER16E"
200 REM -------
210 REM M A I N
220 REM -------
230 GOSUB 1000
240 GOSUB 2500
250 GOSUB 4500
260 GOSUB 2000:X=2:Y=14:GOSUB 4160:PRINT"QUIERE VER  LA RECTA DE REGRESION ?"
270 X=14:Y=16:GOSUB 4160:PRINT"( S o N )
280 A$=INKEY$:IF A$="N" OR A$="n" THEN 310
290 IF A$<>"s" AND A$<>"S" THEN 280
300 GOSUB 5000
310 GOSUB 2000:X=9:Y=14:GOSUB 4160:PRINT"OTRO GRAFICO ? (S o N)"
320 A$=INKEY$:IF A$="N" OR A$="n"THEN GOSUB2000:PLAYBP$+BP$:LOCATE12,13:PRINT"FIN PROGRAMA":END
330 IF A$<>"s" AND A$<>"S" THEN 320
340 FOR I=1 TO MR:Y(I)=0:X(I)=0:X1(I)=0:Y1(I)=0:NEXT:MX=SW(7)
350 IF KV=1 THEN KV=0:GOTO 230
360 GOTO 240
1000 REM ---------------
1010 REM LECTURA FICHERO
1020 REM ---------------
1030 GOSUB 2000:X=1:Y=12:GOSUB 4160:PRINT"1) Los datos estan en la cinta."
1040 Y=14:GOSUB4160:PRINT"2) Los datos se introduciran por video."
1050 X=7:Y=17:GOSUB 4160:PRINT"CUAL ? _"
1060 A$=INKEY$:IF A$<>"1" AND A$<>"2" THEN 1060
1070 X=X+7:GOSUB 4160:PRINT A$:
1080 IF A$="2" THEN GOSUB 3500:RETURN
1090 X=1:Y=21:GOSUB 4160:INPUT"* NOMBRE DEL FICHERO * ";NF$
1100 GOSUB 2000:X=7:Y=12:GOSUB 4160:PRINT"El fichero ha sido generado";CHR$(31);"           por l
a .BASE DE DATOS/ ? (S o N)"
1110 A$=INKEY$:IF A$="S" OR A$="s" THEN GOSUB 2200:IO=2:SW(1)=1:P7=2:GOTO 1140
1120 IF A$<>"n" AND A$<>"N" THEN 1110
1130 GOSUB 1500:SW(1)=0
1140 IF MX>MR THEN GOSUB 4000
1150 REM --- lee cinta ---
1160 GOSUB 2000:X=7:Y=10:GOSUB 4160:PRINT"Posicionar la cinta DATOS":PLAYBP$
1170 X=12:Y=13:GOSUB 4160:PRINT"y apretar PLAY.":X=10:Y=20:GOSUB4160:PRINT"Despues pulsar una te
cla"
1180 IF INKEY$="" THEN 1180
1190 GOSUB 1300:FOR I=1 TO 300:NEXT
1200 OPEN"CAS:"+NF$ FORINPUTAS#1
1210 IF SW(1)=1 THEN INPUT#1,A$:MX=VAL(A$):IF MX>MR THEN GOSUB 4000:GOTO 1240 ELSE GOTO 1240
1220 IF IO=1 THEN 1240
1230 FOR I=1 TO IO-1:INPUT#1,A$:NEXT:A$=""
1240 A=0:K1=1
1250 INPUT#1,B$(K1)
1260 IF SW(3)=3 AND K1<MX THEN FOR I=1 TO SW(4)-1:INPUT#1,A$:NEXT I:A$="":MX=MX-I+1
1270 IF SW(3)=4 THEN A=A+1:IF A=SW(4)-1 AND K1<MX THEN INPUT#1,A$:A$="":A=0:MX=MX-1
1280 K1=K1+1:IF K1<=MX THEN 1250
1290 CLOSE#1:SW(7)=MX:RETURN
1300 REM -------
1310 REM ESPERAR
1320 REM -------
1330 GOSUB 2000:X=14:Y=15:GOSUB 4160:PRINT"E S P E R A R":RETURN
1500 REM --- introduccion formato ---
1510 GOSUB 2000:X=1:Y=10:GOSUB 4160:PRINT"INTRODUCIR NUMERO DEL 1";CHR$(&HF8);" REGISTRO UTIL":
X=14:Y=14:GOSUB 4160:INPUT IO:IF IO<1 THEN 1510
1520 GOSUB 2000:X=5:Y=9:GOSUB 4160:PRINT"INTRODUCIR EL FORMATO REGISTRO:"
1530 X=3:Y=18:GOSUB 4160:INPUT"BYTE principio";P7:IF P7<1 THEN 1530
1540 FOR I=1 TO M3
1550 X=0:FOR Y=11 TO 23:GOSUB 4160:PRINTSPACE$(39);:NEXT Y
1560 X=13:Y=14:GOSUB 4160:PRINT"CAMPO n.";I
1570 Y=15:GOSUB 4160:PRINT"         "
1580 X=1:Y=18:GOSUB4160:INPUT"TIPO (3=alf. 1=num. -1=FIN )";TP(I):IF TP(I)=-1 THEN 1610 ELSE IF
TP(I)<>1 AND TP(I)<>3 THEN 1580
1590 X=9:Y=20:GOSUB 4160:INPUT"LONGITUD";LC(I):IF LC(I)<1 THEN 1590
1600 NEXT
1610 M3=I-1
1620 GOSUB 2000:X=0:Y=13:GOSUB 4160:INPUT"CUANTOS REGISTROS CONTIENE EL FICHERO";MX:MX=MX+1-IO
1630 RETURN
2000 REM --- cabecera ---
2010 CLS:X=0:Y=0:GOSUB 4160:PRINTH1$
2020 X=11:Y=4:GOSUB 4160:PRINT"┌─────────────────┐"
2030 Y=5:GOSUB 4160:PRINT"| G R A F I C O S |"
2040 Y=6:GOSUB 4160:PRINT"└─────────────────┘"
2050 RETURN
2200 REM ---------------
2210 REM lectura FORMATO
2220 REM ---------------
2230 CLS:LOCATE 3,8:PRINT"Rebobinar la cinta DEFINICIONES":LOCATE11,11:PRINT"y apretar PLAY.":LO
CATE8,20:PRINT"Despues pulsar una tecla."
2240 IF INKEY$="" THEN 2240
2250 GOSUB 1300
2260 OPEN"CAS:DEFC"FORINPUTAS#1
2270 FOR I=1 TO M3
2280 INPUT#1,R$
2290 XD(I)=VAL(MID$(R$,1,2))
2300 YD(I)=VAL(MID$(R$,3,2))
2310 D$(I)=MID$(R$,5,10)
2320 XC(I)=VAL(MID$(R$,15,2))
2330 YC(I)=VAL(MID$(R$,17,2))
2340 LC(I)=VAL(MID$(R$,19,2))
2350 TP(I)=VAL(MID$(R$,21,2))
2360 NEXT I:CLOSE#1
2370 RETURN
2500 REM ----------------------
2510 REM ELECCION DE LOS CAMPOS
2520 REM ----------------------
2530 GOSUB 5200
2540 IF KV=1 THEN RETURN
2550 CLS:X=0:Y=0:GOSUB 4160:PRINTH1$
2560 IF SW(1)=0 THEN 2610
2570 FOR I=1 TO M3: X=XD(I):Y=YD(I):GOSUB 4160:PRINT D$(I):NEXT
2580 FOR I=1 TO M3:IF TP(I)=0 THEN 2600
2590 X=XC(I):Y=YC(I):GOSUB 4160:PRINT STR$(I)
2600 NEXT I:GOTO 2630
2610 X=1:FORY=1TOM3:GOSUB4160:PRINT"CAMPO n.";Y;TAB(15)"TIPO ";TP(Y);TAB(25)"LONGITUD ";LC(Y):NE
XT Y
2620 X=0:Y=22:GOSUB 4160:PRINT SPACE$(39);
2630 X=0:Y=22:GOSUB 4160:INPUT"NUMERO CAMPO 'X'(N=progres.)";X$:A=VAL(X$)
2640 IF X$="n" OR X$="N" THEN 2670
2650 IF A<1 OR A>M3 THEN 2620
2660 IFTP(A)=0ORTP(A)=3THEN2620
2670 X=0:Y=22:GOSUB4160:PRINTSPACE$(39);
2680 X=0:Y=22:GOSUB 4160:INPUT"NUMERO CAMPO 'Y'  ";Y$:A=VAL(Y$):IF A<1 OR A>M3 THEN 2670
2690 IFTP(A)=0ORTP(A)=3ORY$=X$THEN2670
2700 X=0:Y=22:GOSUB 4160:PRINT"COORD.'X'=CAMPO ";X$;"   COORD.'Y'=CAMPO ";Y$
2710 X=12:Y=23:GOSUB 4160:PRINT"SEGURO (S o N)";
2720 A$=INKEY$:IF A$="N" OR A$="n" THEN 2500
2730 IF A$<>"s" AND A$<>"S" THEN 2720
2740 KX=VAL(X$):KY=VAL(Y$)
2750 FORJ=1TO2:IFJ=2THENA=KY:GOTO2770
2760 IF KX=0 THEN 2800 ELSE A=KX
2770 P=P7:FOR I=1 TO A
2780 L=LC(I):P=P+L:NEXTI:P=P-L
2790 IFJ=1THENP1=P:L1=L ELSEP2=P:L2=L
2800 NEXT J
2810 FOR I=1 TO MX
2820 A2$=MID$(B$(I),P2,L2):Y(I)=VAL(A2$)
2830 IFKX=0THENX(I)=I:GOTO2850
2840 A1$=MID$(B$(I),P1,L1):X(I)=VAL(A1$)
2850 NEXT I
```

```
2860 IF MX>200 THEN GOSUB 3000:REM --- AGREGACION ---
2870 RETURN
3000 REM ----------
3010 REM AGREGACION
3020 REM ----------
3030 GOSUB 2000:PLAYBP$
3040 X=1:Y=12:GOSUB 4160:INPUT"NUMERO VALORES EJE 'X'(max 200)";N1:IF N1> 200 OR N1<1 THEN 3000
3050 K=MX/N1
3060 I=0:FOR Z=1 TO MX STEP K:I=I+1
3070 S1=0:S2=0:K1=0
3080 FOR V=Z TO Z+K
3090 S1=S1+X(V):S2=S2+Y(V):K1=K1+1
3100 NEXT V
3110 X(I)=S1/K1:Y(I)=S2/K1
3120 NEXT Z
3130 FOR J=I+1 TO MX:X(J)=0:Y(J)=0:NEXT
3140 MX=I:RETURN
3500 REM ----------------------
3510 REM lectura datos de video
3520 REM ----------------------
3530 KV=1:MX=0:GOSUB 2000
3540 X=1:Y=10:GOSUB 4160:PRINT"Cuantos datos desea introd.? (max 130)"
3550 X=13:Y=13:GOSUB4160:INPUTMX
3560 IF MX>130 OR MX<2 THEN 3500
3570 X=4:Y=15:GOSUB 4160:PRINT"1) X = valor progresivo"
3580 X=4:Y=17:GOSUB 4160:PRINT"2) X = valor dato de teclado"
3590 X=11:Y=20:GOSUB 4160:PRINT"CUAL ? _";
3600 A$=INKEY$:IF A$<>"1" AND A$<>"2" THEN 3600
3610 FOR I=1 TO MX:X=0:FOR Y=10 TO 21:GOSUB 4160:PRINT SPACE$(39);:NEXT Y
3620 X=15:Y=11:GOSUB 4160:PRINT"DATO n.";RIGHT$(STR$(I),3)
3630 Y=12:GOSUB 4160:PRINT"----------"
3640 IF A$="1" THEN X(I)=I-1:GOTO 3660
3650 X=2:Y=16:GOSUB 4160:INPUT"introducir el valor de la X ";X(I):IF X(I)<0 THEN 3650
3660 X=2:Y=18:GOSUB 4160:INPUT"introducir el valor de la Y ";Y(I):IF Y(I)<0 THEN 3660
3670 NEXT I:RETURN
4000 REM ---------
4010 REM seleccion
4020 REM ---------
4030 A=MX:A6#=A/MR:A3=INT(A/(A-MR)):IF A6#<>INT(A6#) THEN A2=INT(A6#)+1 ELSE A2=INT(A6#)
4040 GOSUB 2000:X=3:Y=9:GOSUB4160:PRINT"EL NUMERO DE REGISTROS ES SUPERIOR";CHR$(31);"    AL MAX
IMO ACEPTABLE (";MR;")"
4050 X=5:Y=15:GOSUB4160:PRINT"ELIJA EL CRITERIO DE SELECCION:"
4060 X=1:Y=17:GOSUB 4160:PRINT"1) Descartar los primeros ";A-MR;" registros utiles"
4070 Y=18:GOSUB 4160:PRINT"2) Descartar los ultimos ";A-MR;"registros"
4080 Y=19:GOSUB 4160:PRINT"3) Leer un registro cada ";A2
4090 A$="3":IF A3>1 THENY=20:GOSUB 4160:PRINT"4) Descartar un registro cada ";A3:A$="4"
4100 B$=INKEY$:IF B$<"1" OR B$>A$ THEN 4100
4110 IF B$="1" THEN I0=I0+A-MR:MX=MR
4120 IF B$="2" THEN MX=MR
4130 SW(3)=VAL(B$):IF SW(3)=3 THEN SW(4)=A2
4140 IF SW(3)=4 THEN SW(4)=A3
4150 RETURN
4160 REM --- posiciona cursor ---
4170 LOCATEX,Y:RETURN
4500 REM --------
4510 REM graficos
4520 REM --------
4530 GOSUB 2000:X=11:Y=12:GOSUB 4160:PRINT"1) HISTOGRAMA"
4540 Y=15:GOSUB 4160:PRINT"2) GRAFICO":X=14:Y=20:GOSUB 4160:PRINT"cual ? _"
4550 A$=INKEY$:IFA$<>"1"ANDA$<>"2"THEN4550
4560 GOSUB 2000:X=11:Y=10:GOSUB4160:PRINT"ELIJA EL COLOR:"
4570 X=15:Y=14:GOSUB 4160:PRINT"1) NEGRO"
4580 Y=16:GOSUB 4160:PRINT"2) BLANCO"
4590 Y=18:GOSUB 4160:PRINT"3) ROJO"
4600 Y=20:GOSUB 4160:PRINT"4) AZUL"
4610 C$=INKEY$:IFC$<"1"ORC$>"4"ORC$=""THEN4610
4620 CC=VAL(C$):IFCC=2THENCC=15ELSEIFCC=3THENCC=6
4630 IFA$="2"THENSW(2)=1:GOTO4650
4640 SW(2)=0:IFMX>14THENN1=14:GOSUB1300:GOSUB3050
4650 XM=X(1):YM=Y(1)
4660 FORI=2TOMX:IFX(I)>XMTHENXM=X(I)
4670 IFY(I)>YMTHENYM=Y(I)
4680 NEXTI
4684 IF YM=0 THEN YM=1E-05
4686 IF XM=0 THEN XM=1E-05
4690 FOR I=1 TO MX
4700 X(I)=24+X(I)*180/XM
4710 Y(I)=15+Y(I)*155/YM:Y(I)=185-Y(I)
4720 NEXT
4730 SCREEN2:COLOR1,14,5:CLS
4740 OPEN"GRP:"FOROUTPUTAS#2
4750 A=171:LINE(23,15)-(23,A),15
4760 D=0:IFSW(2)=0THEND=10
4770 LINE(23,A)-(205+D,A),15
4780 LINE(19,15)-(22,15),15
4790 LINE(19,A+2)-(22,A),15
4800 PRESET(10,173):PRINT#2,"0";
4810 PRESET(0,6):PRINT#2,INT(YM+.5);
4820 COLOR12:PRESET(90,1):PRINT#2,"Ediciones Forum,S.A."
4830 PRESET(30,184):PRINT#2,"E.G.S.            11/85":COLOR1
4840 IF SW(2)=0 THEN GOSUB 4920:GOTO 4970
4850 LINE(205,A+1)-(205,A+3),15
4860 XM$=STR$(INT(XM+.5)):L=LEN(XM$):A=190:IF L>8 THEN A=A-(L-8)*8
4870 PRESET(A,176):PRINT#2,XM$
4880 FOR I=1 TO MX-1
4890 LINE(X(I),Y(I))-(X(I+1),Y(I+1)),CC
4900 NEXT:COLOR15,4
4910 GOTO4970
4920 REM --- histograma ---
4930 L=7*14/MX
4940 A=24:FOR I=1 TO MX
4950 LINE(A,Y(I))-(A+L,170),CC,BF
4960 A=A+L+L:NEXT:RETURN
4970 IF INKEY$="" THEN 4970
4980 CLOSE#2:SCREEN0:COLOR15,4:IFSW(2)=0 THEN GOSUB 5300
4990 RETURN
5000 REM ------------------
5010 REM RECTA DE REGRESION
5020 REM ------------------
5030 N=MX:A1=0:A2=0:A3=0:A4=0
5040 FOR I=1 TO N
5050 A1=A1+X1(I):A2=A2+Y1(I):A4=A4+X1(I)*Y1(I):A3=A3+X1(I)^2
5060 NEXT I
5070 D=N*A3-A1^2:IFD=0THEN5152
5075 B=(A2*A3-A1*A4)/D
5080 A=(N*A4-A1*A2)/D
5085 IF A<.1 AND A>-.1 THEN 5152
5090 X0=-B/A:Y0=B
5100 GOSUB 2000:X=0:Y=10:GOSUB4160:PRINT"LA ECUACION DE LA RECTA DE REGRESION";CHR$(31);"     ES:
  Y =";:A$=STR$(INT(A)):L=LEN(A$):A$=STRING$(L,"#")+".#":PRINTUSINGA$;A;:PRINT" * X";
5110 IFB<>0THENA$=STR$(INT(B)):L=LEN(A$):A$="+"+STRING$(L,"#")+".#":PRINTUSINGA$;B
5120 X=4:Y=16:GOSUB 4160:PRINT"LA RECTA CORTA:"
5130 X=4:Y=19:GOSUB 4160:PRINT"EL EJE X EN EL PUNTO X=";:A$=STRING$(LEN(STR$(INT(X0))),"#")+".#"
:PRINT USING A$;X0
5140 X=4:Y=21:GOSUB4160:PRINT"EL EJE Y EN EL PUNTO Y=";:A$=STRING$(LEN(STR$(INT(Y0))),"#")+".#":
PRINT USING A$;Y0
5150 X=12:Y=23:GOSUB4160:PRINT"pulse una tecla";:GOTO 5160
5152 GOSUB 2000:X=8:Y=12:GOSUB 4160:PRINT"LA ECUACION DE LA RECTA";CHR$(31);"                NO
 ES REPRESENTABLE"
5155 X=12:Y=20:GOSUB 4160:PRINT"Pulse una tecla":PLAYBP$+BP$
5160 IF INKEY$="" THEN 5160
5170 RETURN
5200 REM ------------------------
5210 REM SALVA VALORES ORIGINALES
5220 REM ------------------------
5230 FOR I=1 TO MX:X1(I)=X(I):Y1(I)=Y(I)
5240 NEXT:RETURN
5300 REM --- IMPRESION HISTOGRAMA ---
5310 GOSUB 2000:X=1:Y=12:GOSUB 4160:PRINT"SE DESEA LA IMPRESION DEL HISTOGRAMA ?"
5320 X=15:Y=15:GOSUB 4160:PRINT"(S o N)"
5330 A$=INKEY$:IF A$="N" OR A$="n" THEN RETURN
5340 IF A$<>"s" AND A$<>"S" THEN 5330
5350 X=0:Y=18:GOSUB 4160:PRINT"POSICIONAR EL PAPEL Y PULSAR UNA TECLA":PLAYBP$
5360 IF INKEY$="" THEN 5360
5370 GOSUB 2000:X=4:Y=10:GOSUB 4160:PRINT"INTRODUCIR LA CABECERA DE IMPRESION"
5380 X=0:Y=13:GOSUB 4160:INPUTA$
5390 LPRINT CHR$(14);H1$;CHR$(15):LPRINT:LPRINT:LPRINTTAB((80-LEN(A$))/2);A$:LPRINT:LPRINT:LPRIN
TCHR$(27);CHR$(66):LPRINT" |":LPRINT" |"
5400 FOR I=1 TO MX
5405 IF Y(I)>170 THEN Y(I)=170
5410 LPRINT" |";STRING$((170-Y(I))/3,"█")
5420 LPRINT" |";STRING$((170-Y(I))/3,"█")
5430 LPRINT" |":LPRINT" |":NEXT
5440 LPRINT TAB(52);"MAX=";YM
5450 LPRINT CHR$(27);CHR$(65):RETURN
```

El tratamiento de textos

La posibilidad de preparar textos de manera rápida, con facilidad de corrección, es una necesidad cada vez más apremiante en cualquier actividad. El uso de la máquina de escribir impone muchas limitaciones, como la dificultad de las correcciones (la mayoría de veces, la sustitución de una sola palabra implica rehacer toda una página), imposibilidad de obtener copias al cabo de un tiempo, sustituir o insertar párrafos y muchas otras.

Todas estas limitaciones desaparecen utilizando un ordenador. Con éste, un texto cualquiera puede memorizarse y, por tanto, variarse como se desee y volverse a imprimir cada vez que sea necesario. Además, la posibilidad de utilizar datos procedentes de otros programas permite reunir en una impresión única textos distintos, confiriendo a la aplicación una elasticidad de empleo que es absolutamente imposible con la máquina de escribir.

El programa presentado permite transformar el propio ordenador en una máquina de escribir con memoria, capaz de ofrecer prestaciones análogas a las de dispositivos mucho más caros.

Uso del programa

El programa para el tratamiento de textos permite utilizar el ordenador como una máquina de escribir con todas las posibilidades de corrección, búsqueda y sustitución propias de un ordenador.

La estructura básica del software es muy sencilla: se trata de una versión diferente de la data entry, potenciada o implantada con algunas funciones nuevas. Las principales son:

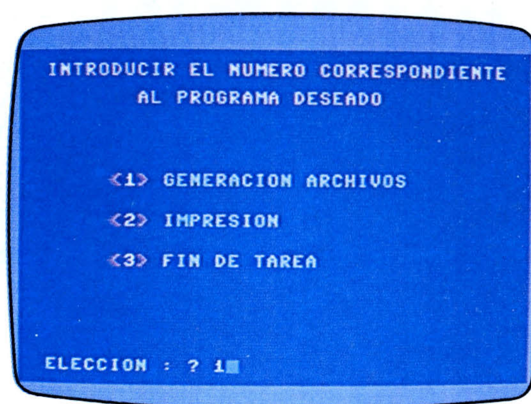
- Posibilidad de escribir en toda la pantalla y de enviar a impresión con líneas de 80 columnas (la pantalla prevé 40)
- Memorización en diskette (o cinta) del texto
- Fusión de un texto con datos memorizados en otro fichero, por ejemplo una lista de direcciones

Por tanto, la data entry debe modificarse como sigue:

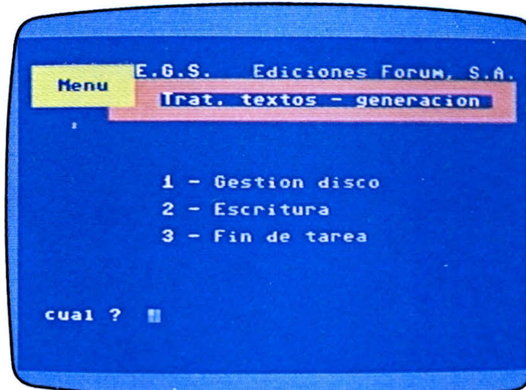
- Todos los campos son alfanuméricos y de 40 bytes de longitud
- No hay que utilizar la subrutina de presentación de máscara
- El texto debe memorizarse en un buffer separado y transferido a B$(*) sólo para la parte correspondiente a la pantalla durante el proceso

Finalmente, a causa de la diferencia de anchura entre la pantalla y la impresora, hay que insertar un código de control (tecla RETURN) que indique cuándo se pide un cambio de línea, por lo que la tecla RETURN sólo debe utilizarse cuando se quiere que la impresora vaya «al principio» y debe incluirse en el buffer (la data entry de la primera versión descarta un código así). Para poner de manifiesto la presencia del comando RETURN (al principio), en algunas versiones aparece en la pantalla el signo "<". Durante las primeras fases de utilización del programa deberá prestarse mucha atención a este símbolo, porque cuando el sistema lo encuentra, realiza la unión de dos líneas de 40 (de pantalla) para formar la cadena de impresión de 80. También deberá prestarse atención en la escritura de líneas con palabras cortadas, porque la función de justificación puede producir diferencias con respecto al resultado deseado.

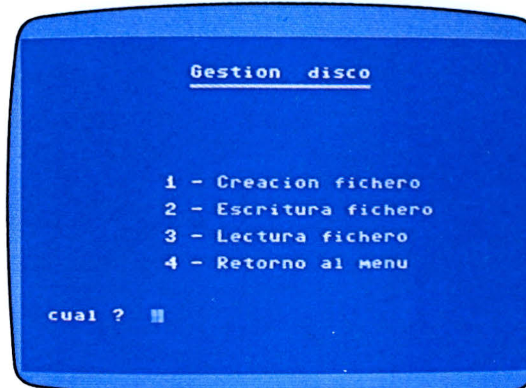
Menú del programa de escritura de textos. Tecleando el número correspondiente (1, 2, 3) se activará la función deseada.



Main. El procedimiento Tratamiento de Textos en la versión C-64 está compuesto por dos programas separados, activados por el módulo cargador. Con el primero es posible escribir y grabar los textos en el disco de datos.



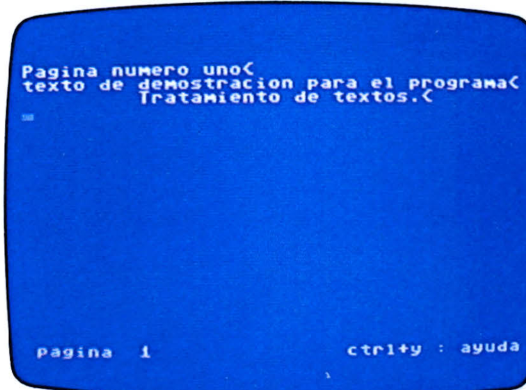
Gestión de disco. Esta función está compuesta por varias opciones, por lo que existe un menú secundario: la voz 2 permite grabar en el disco de datos el texto que hay en la memoria; la 3 permite cargar un texto grabado anteriormente en el disco; tecleando la 4 se puede volver al menú principal (foto anterior).



Creación del fichero datos. Del menú secundario Gestión de disco se ha activado la voz 1 (Creación fichero): la primera petición del programa es el nombre del fichero datos en el que se grabará el texto. A continuación habrá que implantar el número de páginas y el de líneas por página deseados.



Escritura. Se ha activado la voz 2 del menú principal: se dispone ahora de la pantalla vídeo para la redacción del texto. En esta fase, los comandos disponibles son:
F1 = salta a la página siguiente
F2 = salta a la página anterior
F3 = inserta un carácter
F4 = borra un carácter
F5 = inserta una línea
F6 = borra una línea
F7 = busca una palabra
F8 = introduce y vuelve al menú
Para el desplazamiento del cursor, las teclas son:
CRSR UP = arriba
CRSR DOWN = abajo
CRSR LEFT = a la izquierda
CRSR RIGHT = a la derecha



Durante la escritura del texto, la última línea de la pantalla contiene mensajes útiles para el usuario, como la indicación de la página actual y de la función de ayuda.

Inserción. Esta función permite la inserción de un carácter cada vez en la posición en que se encuentra el cursor; los caracteres que hay a la derecha de éste se desplazan automáticamente.

Búsqueda de una palabra. Se ha activado la tecla F7 y el programa espera la introducción (última línea de la parte inferior de la pantalla) de la palabra a identificar en el texto que hay en pantalla.
Una vez hallada la palabra pedida (máx. 15 caracteres), el cursor se posicionará encima de ella.

Inserción de una línea. El cursor está en la línea 6 y se activa el comando mediante la tecla F5.

Entre las líneas cuatro y seis se ha insertado una nueva línea: todo el texto que hay debajo del cursor se desplaza hacia abajo y, eventualmente, una parte del mismo pasa a la página siguiente.

```
Ejemplo de insercion de linea:<
linea uno<
linea dos<
linea tres<
linea cuatro<
linea cinco
linea seis<
linea siete<
linea ocho<
linea nueve<
linea diez<

pagina  1                    ctrl+y : ayuda
```

En el texto anterior se activará el comando de borrado de línea (tecla F6). El cursor está en la línea ocho.

```
Ejemplo de cancelacion de linea:<
linea uno<
linea dos<
linea tres<
linea cuatro<
linea cinco<
linea seis<
linea siete<
linea ocho<
linea nueve<
linea diez<

pagina  1                    ctrl+y : ayuda
```

Con el comando de borrado de línea, el texto se traslada una posición hacia arriba y en la memoria se borra el texto que había en la línea borrada.

```
Ejemplo de cancelacion de linea:<
linea uno<
linea dos<
linea tres<
linea cuatro<
linea cinco<
linea seis<
linea siete<
linea diez<

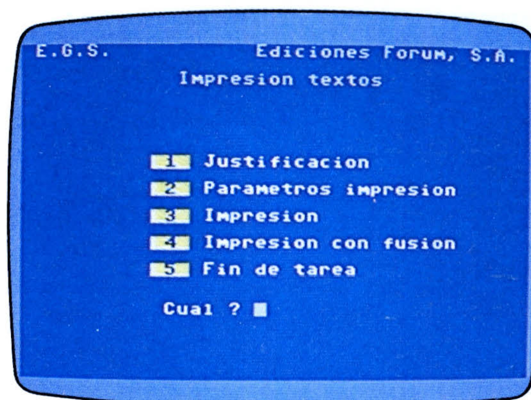
pagina  1                    ctrl+y : ayuda
```

Programa de impresión. Apenas activado este programa, se pide al usuario el nombre del fichero datos (máx. 6 caracteres) en el que hay grabado el texto a imprimir.

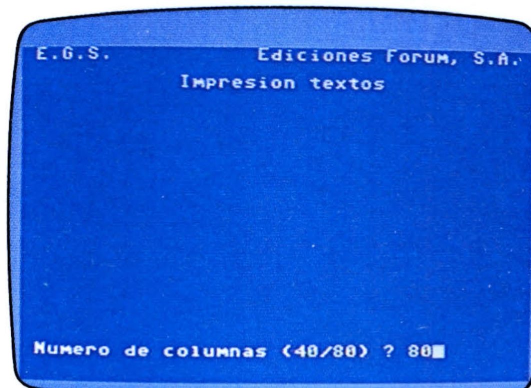
```
E.G.S.              Ediciones Forum, S.A.
            Impresion textos

Nombre del fichero? texto
```

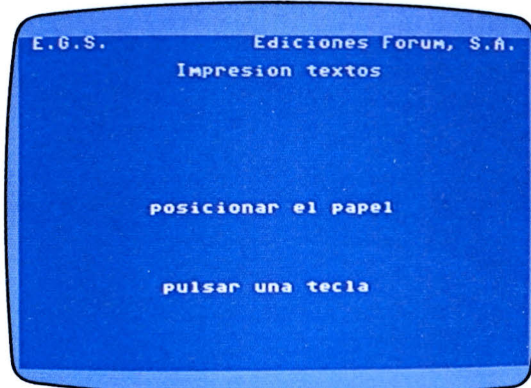
Una vez leído e introducido en memoria el texto a imprimir, se pasa al menú principal del programa, que permite activar la función deseada entre las indicadas en esta foto.



Justificación. Al activar esta función es necesario introducir los parámetros de impresión, o sea: número de columnas (40/80), número de líneas por página y si se desea la numeración de las páginas o no.



Impresión. Antes de pasar a la impresión, el programa avisa al usuario que debe colocar el papel de acuerdo con el modo deseado. La impresión empezará con la pulsación de una tecla.



Impresión con fusión. Esta opción permite la impresión de un texto unido a los datos preparados con la Base de Datos; para ello hay que leer del correspondiente disco los parámetros de la máscara creada con el programa de archivado (Base de Datos).



La lógica del programa

Abajo se ha representado el diagrama general. Las funciones realizadas por las subrutinas son:

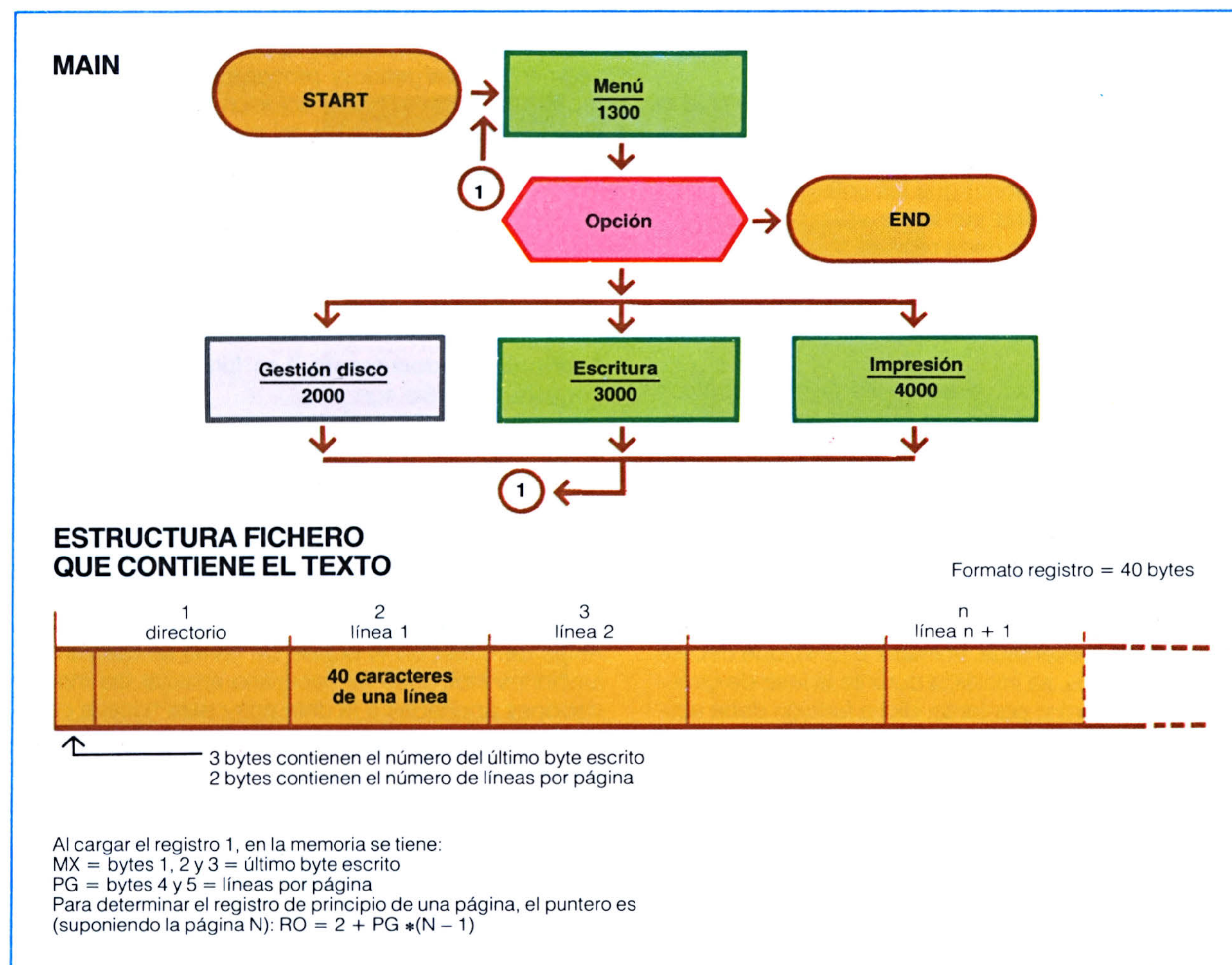
- Gestión del disco: permite la memorización de un documento y su carga en la memoria
- Escritura: activa la data entry para la introducción del texto
- Impresión: permite la impresión del documento que hay en la memoria

En la misma figura se ha representado la estructura de fichero utilizada que se prevé en un registro de 80 bytes, o sea una línea de impresión. Además, el programa permite una estructuración por página de longitud, en número de líneas, parametrizada. Con la activación del programa, la primera función a realizar es la creacióan del fichero que deberá contener el documento (mediante la subrutina 2000). Si no, el texto introducido en sucesión se perderá. Sin embargo, el programa puede utilizarse sin fichero.

En la fase de creación se pide la longitud máxima del texto en número de páginas, que no es necesario responder de manera exacta, sino que basta un valor aproximado por exceso.
Terminada esta fase, puede activarse la introducción del texto, o sea la subrutina de escritura (3000) y después la de impresión (4000). Finalmente, si se desea memorizar el documento, hay que llamar nuevamente la gestión de disco.
Para terminar, hay que tener en cuenta que en el diagrama no se ha indicado la subrutina de inicialización que, en cambio, está en los listados y que algunas subrutinas utilizadas a continuación son modificaciones de las que ya se han presentado, por lo que no se da su descripción.

Gestión disco (subrutina 2000). Las funciones previstas en esta subrutina son:

1 / Creación del fichero



2 / Memorización de un documento
3 / Transferencia a la memoria de un documento memorizado

Las tres funciones están incluidas en un menú secundario que se presenta activando la 2000.
Como se ha indicado anteriormente, la cantidad de datos (o sea de las líneas de texto) que hay en la memoria es superior a las 20 indicadas en la pantalla, por lo que hay que utilizar dos buffers:

B$(20) = buffer vídeo, que contiene las líneas (de vídeo) presentadas y permite la modificación (o la introducción) vía la data entry
B1$(*) = buffer total, contiene todo el documento

Por tanto, las operaciones de I/O con el disco deben gestionar el segundo buffer (B1$(*)).
En la página siguiente se ha representado el diagrama de la subrutina 2000: como puede verse, está constituida esencialmente por un menú secundario que, en función de la opción elegida, activa las funciones:

- Generación fichero
- Escritura fichero
- Lectura fichero

Obsérvese que, en esta versión, el formato de registro es fijo a 80 caracteres, lo que significa que si algunas líneas no son completas, ocupan el mismo espacio (en el disco y en la memoria) que las enteras.
Una alternativa podría ser la de utilizar ficheros secuenciales en los que cada registro ocupa un número de bytes igual a los caracteres contenidos. En este caso se tendría un ahorro de espacio pero, en cambio, los datos no serían direccionables, impidiendo así toda implantación.
Como se ha indicado, si se desea la memorización sucesiva de los datos, hay que activar en primer lugar la generación del fichero. Para esta función, el programa pide al usuario la introducción de:

1 / Nombre del fichero, en la variable de cadena NM$
2 / Número de páginas previstas en NP
3 / Líneas por página (en impresión) en la variable PG

En la versión presentada, la matriz B1$(*), que deberá contener el texto, se inicializa durante la fase de generación del fichero y, por tanto, dicha función debe activarse forzosamente, incluso si no se prevé la memorización de los datos. Sin embargo, es posible la extracción de esta subrutina para no ligar las dos funciones: generación del fichero e inicialización del buffer. En el listado propuesto se ha preferido esta solución porque se presta más fácilmente a implantaciones.
Además, como ya se ha indicado, tanto el fichero como la ocupación de memoria (es decir, la inicialización de B1$(*)) están estructurados para el valor N, igual al máximo previsto. Esto implica una mayor ocupación de memoria que lo estrictamente necesario, pero ofrece una mayor elasticidad porque contempla la utilización de ficheros secuenciales, solución que siempre puede adoptarse con pocas modificaciones.
Finalmente, obsérvese que el diagnóstico obtenido comprobando que el número de registros no supere el valor 998 sólo es formal y no asegura que la memoria disponible pueda realmente contenerlos todos; efectivamente, 998 registros de 40 bytes de longitud ocupan 998 × 40 = 39920 bytes totales, valor muy superior a la capacidad de la memoria.
Para obtener un diagnóstico más exacto hay que modificar el valor límite en base a la capacidad de memoria de la máquina utilizada.
Un método aproximado, pero sin embargo válido, consiste en comprobar cuántos bytes quedan después de haber cargado el programa, calculando con este valor el número máximo de registros (dado por el cociente bytes libres/40). Por ejemplo, con una máquina de 32 Kbytes de memoria central, una vez descontados aproximadamente 18 Kbytes de programa, quedan 14 Kbytes disponibles para datos, lo que equivale a 14000/40 = 175 registros.
Para aumentar este valor, además de la obvia solución de expandir la memoria, se puede trabajar de modo segmentado cargando y procesando grupos de 175 registros cada vez. Esta solución puede obtenerse bien modificando la lógica del programa, bien mucho más sencillamente utilizando varios ficheros para el mismo tema.
Esta ultima solución, prácticamente apta sólo para las versiones disco, no requiere ninguna modificación del programa, puesto que sólo implica un aumento de las instrucciones que debe proporcionar el usuario.

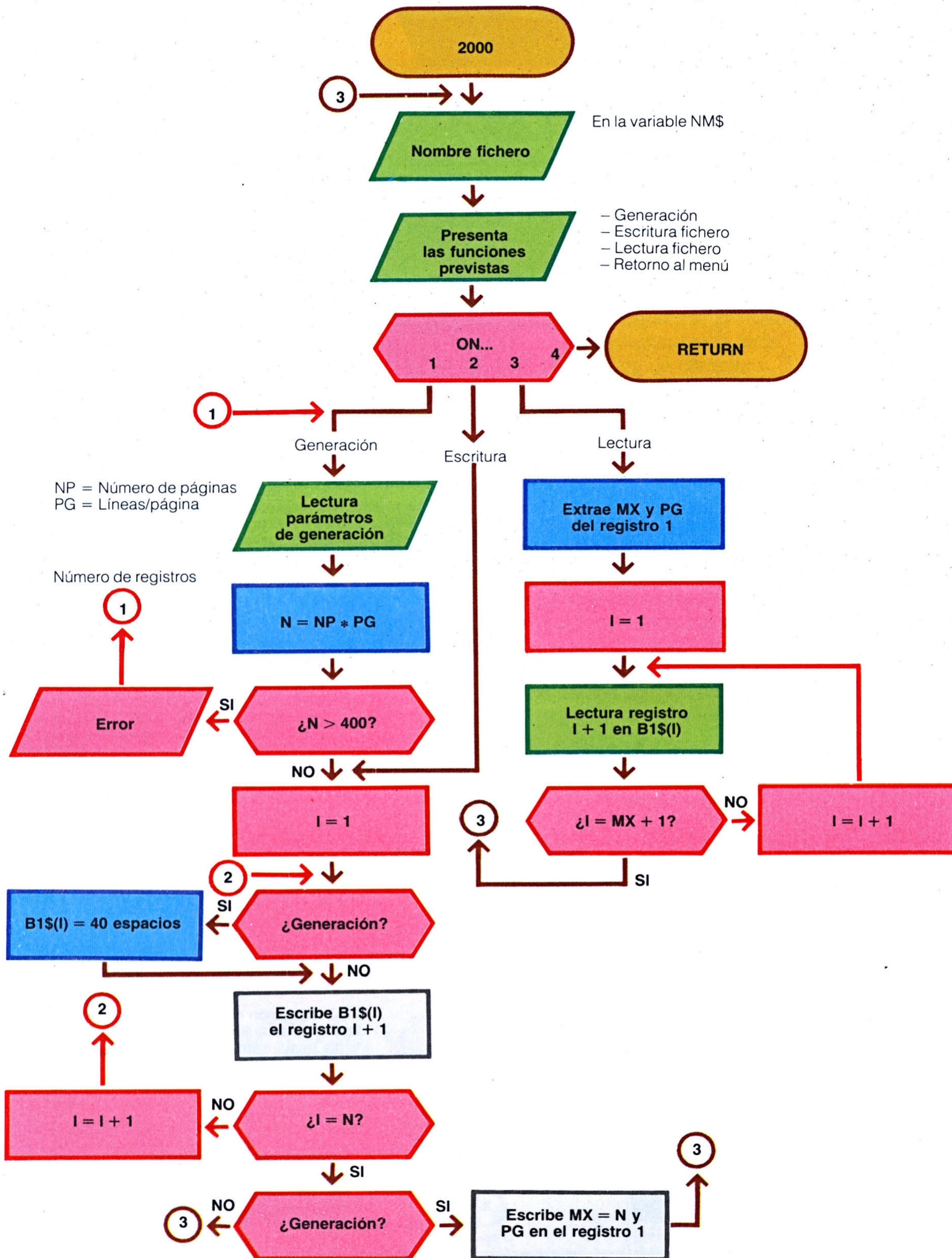
Escritura del documento. Las funciones realizadas por esta subrutina son:

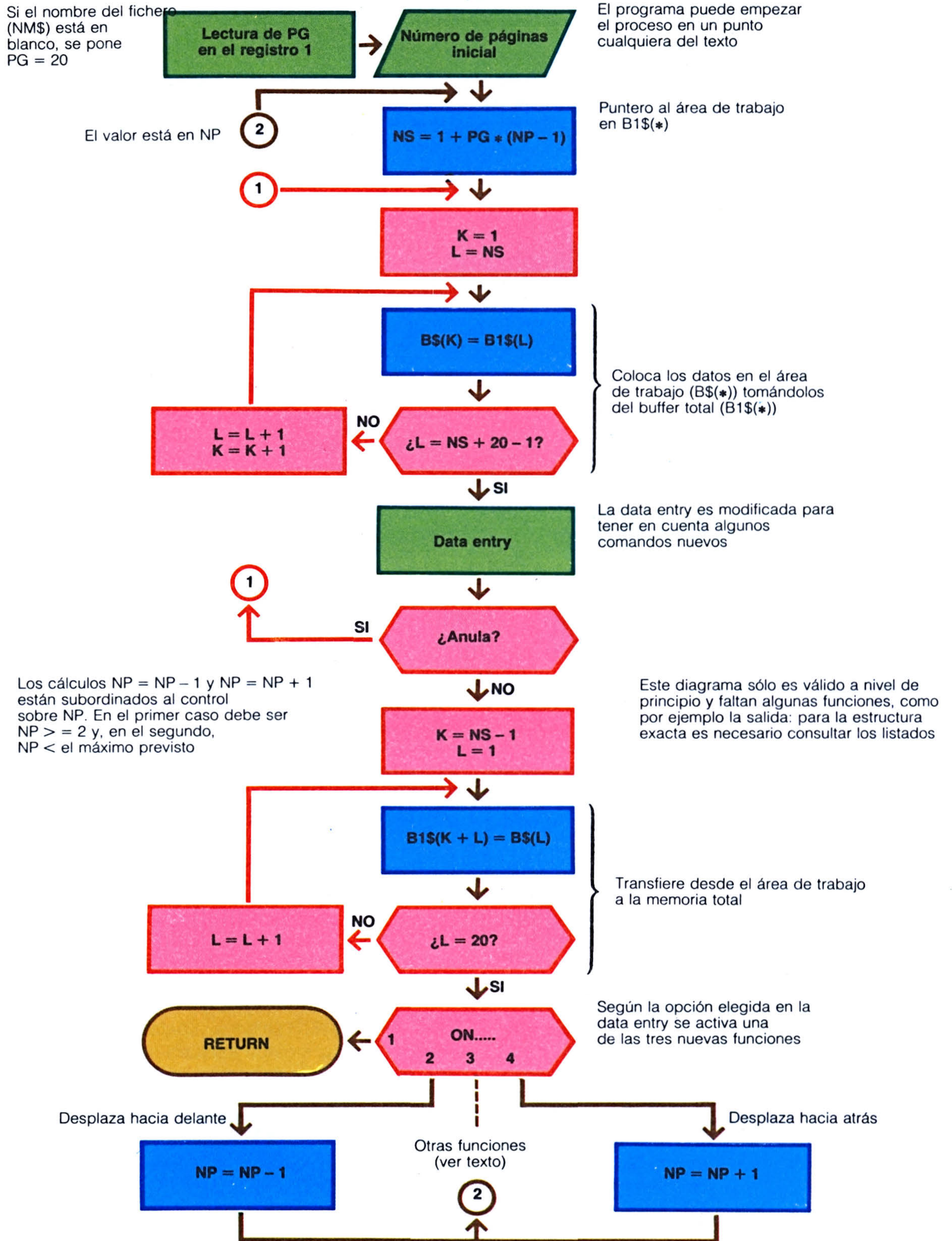
- Selección de la página vídeo a procesar
- Transferencia de los datos correspondientes a la página seleccionada por el buffer total al de trabajo gestionado por la data entry
- Desplazamiento de la página para los siguientes procesados

Antes de entrar en el detalle de cómo se realizan las funciones indicadas es necesario analizar las modificaciones aportadas a la data entry para hacerla compatible con esta aplicación. Las implantaciones necesarias son:

- Nueva lógica de gestión de la tecla RETURN
- Introducción de otras dos teclas funcionales (desplazamiento)

Y la eliminación de las partes que ya no son necesarias como:


SUBRUTINA GESTION DISCO
2000
3
Nombre fichero
En la variable NM$
Presenta las funciones previstas
– Generación
– Escritura fichero
– Lectura fichero
– Retorno al menú
ON...
1 2 3 4
RETURN
1
Generación
Escritura
Lectura
NP = Número de páginas
PG = Líneas/página
Lectura parámetros de generación
Extrae MX y PG del registro 1
Número de registros
1
N = NP * PG
I = 1
Lectura registro I + 1 en B1$(I)
Error
SI
¿N > 400?
NO
I = 1
3
¿I = MX + 1?
NO
I = I + 1
SI
2
B1$(I) = 40 espacios
SI
¿Generación?
NO
2
Escribe B1$(I) el registro I + 1
I = I + 1
NO
¿I = N?
SI
3
NO
3
¿Generación?
SI
Escribe MX = N y PG en el registro 1


ESCRITURA DE UN DOCUMENTO
Si el nombre del fichero (NM$) está en blanco, se pone PG = 20
Lectura de PG en el registro 1
Número de páginas inicial
El programa puede empezar el proceso en un punto cualquiera del texto
El valor está en NP
2
NS = 1 + PG * (NP – 1)
Puntero al área de trabajo en B1$(*)
1
K = 1
L = NS
B$(K) = B1$(L)
Coloca los datos en el área de trabajo (B$(*)) tomándolos del buffer total (B1$(*))
L = L + 1
K = K + 1
NO
¿L = NS + 20 – 1?
SI
Data entry
La data entry es modificada para tener en cuenta algunos comandos nuevos
1
SI
¿Anula?
NO
Los cálculos NP = NP – 1 y NP = NP + 1 están subordinados al control sobre NP. En el primer caso debe ser NP > = 2 y, en el segundo, NP < el máximo previsto
Este diagrama sólo es válido a nivel de principio y faltan algunas funciones, como por ejemplo la salida: para la estructura exacta es necesario consultar los listados
K = NS – 1
L = 1
B1$(K + L) = B$(L)
Transfiere desde el área de trabajo a la memoria total
L = L + 1
NO
¿L = 20?
SI
RETURN
1
ON.....
2 3 4
Según la opción elegida en la data entry se activa una de las tres nuevas funciones
Desplaza hacia delante
Desplaza hacia atrás
NP = NP – 1
Otras funciones (ver texto)
2
NP = NP + 1

- Subrutina de cálculo y su gestión
- Partes correspondientes a la gestión de los campos numéricos
- Alineados

La tecla RETURN, como se ha indicado, debe gestionarse de manera completamente diferente, puesto que ya no debe llamar la subrutina de alineado, sino que debe estar incluida en el buffer para controlar el retorno al principio del carro de la impresora.
En esta gestión de la data entry, los desplazamientos de una línea a otra pierden su significado de «terminadores de campos» y, por tanto, deben efectuarse únicamente con las teclas de desplazamiento. Y viceversa, la tecla RETURN adquiere el significado de comando para la impresora. En cambio, en lo referente a los desplazamientos, se trata de una función completamente nueva, que consiste en presentar partes antecedentes o siguientes a la que está en gestión. Es decir, se trata de activar dos teclas funcionales dedicadas a la transferencia entre el buffer B1$(*) y el de trabajo. Esta transferencia se realiza en grupos de 20 líneas: así se tiene la subdivisión de todo el documento en páginas «lógicas» de 20 líneas cada una (este valor, sin embargo, no tiene nada que ver con el número de líneas por página durante la impresión).
Las funciones y las modificaciones descritas constituyen los principales elementos de esta nueva versión de la data entry, pero no incluyen todas las necesidades que pueden producirse durante la preparación de un texto: por tanto, hay que prever un mayor número de teclas funcionales, cada una dedicada a una actividad específica, indicadas en el diagrama con la línea de trazos (otras funciones).

Como el desarrollo de nuevas funciones requiere la correspondiente presencia de subrutinas, a continuación se relacionan todas las previstas en principio, mientras que las activadas realmente dependen de la capacidad de memoria de la máquina utilizada; además, el enfoque anterior, basado en la modificación de la data entry indicada, no es conveniente porque las modificaciones necesarias hacen más farragosa su adaptación que su nueva escritura completa. En realidad, en los listados se ha seguido este segundo procedimiento y todo lo que se ha indicado anteriormente debe considerarse sólo como una posibilidad.
Las funciones a incluir en la data entry son:

- Comando de transferencia al buffer total (equivalente al anterior comando de introducción)
- Desplazamiento hacia adelante
- Desplazamiento hacia atrás
- Inserción de un carácter
- Borrado de un carácter
- Inserción de una línea
- Borrado de una línea
- Búsqueda de una palabra
- Ayuda

Además de las teclas normales de desplazamiento del cursor (ver la tabla de abajo), cuyos códigos se memorizan por separado para una mayor claridad, TF(10) contiene los códigos correspondientes a las 9 teclas funcionales (más una disponible), mientras que una nueva matriz, TS(4), se dedica a la realización de los desplazamientos.
En la página siguiente se ha representado el diagrama de esta nueva versión de la data entry.

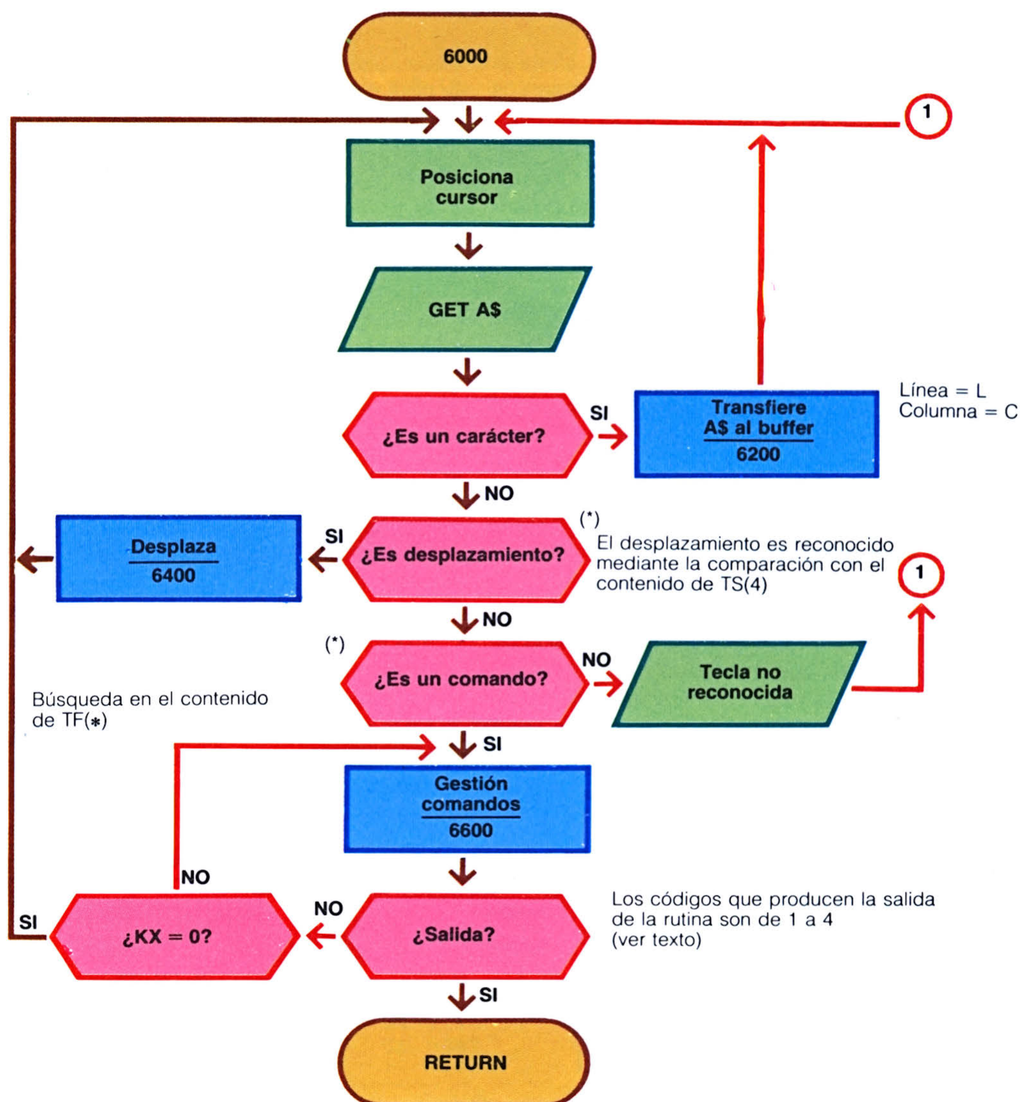
POSICIONES Y CODIGOS UTILIZADOS EN LA DATA ENTRY

Desplazamientos

C-64		MSX		Apple		
Código	Tecla	Código	Tecla	Código	Tecla	Función
145	↑	30	↑	23	CTRL + W	Desplaza arriba
17	↓	31	↓	26	CTRL + Z	Desplaza abajo
29	→	28	→	19	CTRL + S	Desplaza a derecha
157	←	29	←	1	CTRL + A	Desplaza a izquierda

Función

Posición	C-64		MSX		Apple		
en TF(10)	Código	Tecla	Código	Tecla	Código	Tecla	Función
1	140	F8	5	F2	5	CTRL + E	Vuelve a menú + Introducción
2	133	F1	16	F8	16	CTRL + P	Página adelante
3	137	F2	15	F7	15	CTRL + O	Página atrás
4	134	F3	4	F3	4	CTRL + I	Inserta 1 carácter
5	138	F4	9	F4	9	CTRL + D	Borra 1 carácter
6	135	F5	12	F5	12	CTRL + L	Inserta 1 línea
7	139	F6	11	F6	11	CTRL + K	Borra 1 línea
8	136	F7	18	F9	18	CTRL + R	Busca palabra



Data entry. Arriba se ha representado el diagrama de principio. A nivel sintético es análogo al de la data entry precedente y las diferencias residen en las tres subrutinas de gestión (6200, 6400 y 6600).
En particular, se han eliminado los controles sobre el tipo de los campos y sobre las longitudes, puesto que todos los campos son alfanuméricos y de 40 caracteres de longitud. Se han implantado algunas funciones, como la inserción y el borrado de una línea.
Obsérvese que la salida de la subrutina sólo se realiza con una de las 3 primeras teclas funcionales, o sea:

1 / Introducción: el contenido de la pantalla se transfiere al buffer total

2 / Desplazamiento hacia adelante: el contenido de la pantalla es sustituido por las 20 líneas que siguen

3 / Desplazamiento hacia atrás: el contenido de la pantalla es sustituido por las 20 líneas precedentes. Los datos ofrecidos por la primera pantalla se pierden

4 / Salida: el programa vuelve al menú principal.

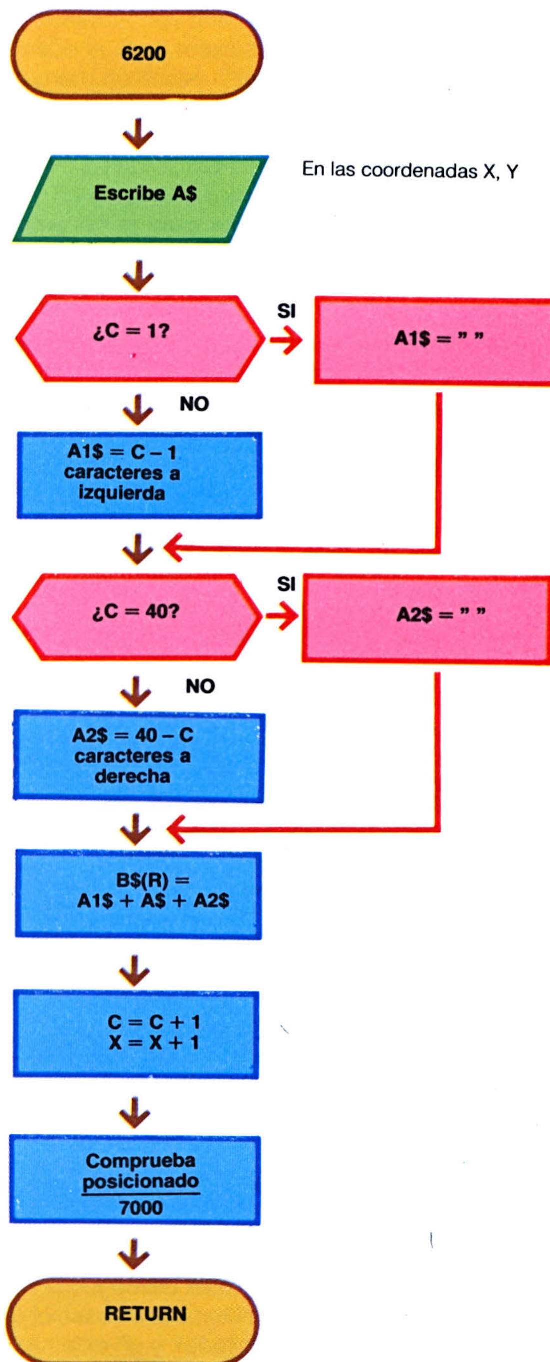
La lógica asociada a estas teclas implica la necesidad de confirmar cada introducción o modificación (tecla 1) antes de continuar; por tanto, antes de activar una cualquiera de las funciones, hay que acordarse de confirmar la introducción mediante la tecla 1.

Transferencia al buffer. El diagrama es similar al utilizado en la data entry anterior y sólo se presenta como indicación.
La diferencia con respecto al otro consiste en el empleo de una longitud de buffer no parametrizada, sino implantada por programa igual a 40 caracteres, que representa la anchura estándar de la pantalla en este tipo de máquina.
Para adecuar el programa en los casos en que se han previsto 80 columnas, basta con sustituir este valor con el que hay en los listados (40).

Comprobación del posicionado. Con cada introducción o desplazamiento del cursor hay que calcular los nuevos valores de L,C y X,Y. Para evitar la repetición en varios puntos del programa de las mismas instrucciones, se ha previsto esta subrutina que, dada la posición actual, calcula la de la futura teniendo en cuenta los posicionados al principio de la línea si la actual está completa y al principio de la pantalla si la línea es la última.

TRANSFERENCIA AL BUFFER

En llamada:
A$ = carácter a insertar
L = contador de línea (identifica el buffer B$(*)
C = contador de carácter
X,Y = posición del cursor



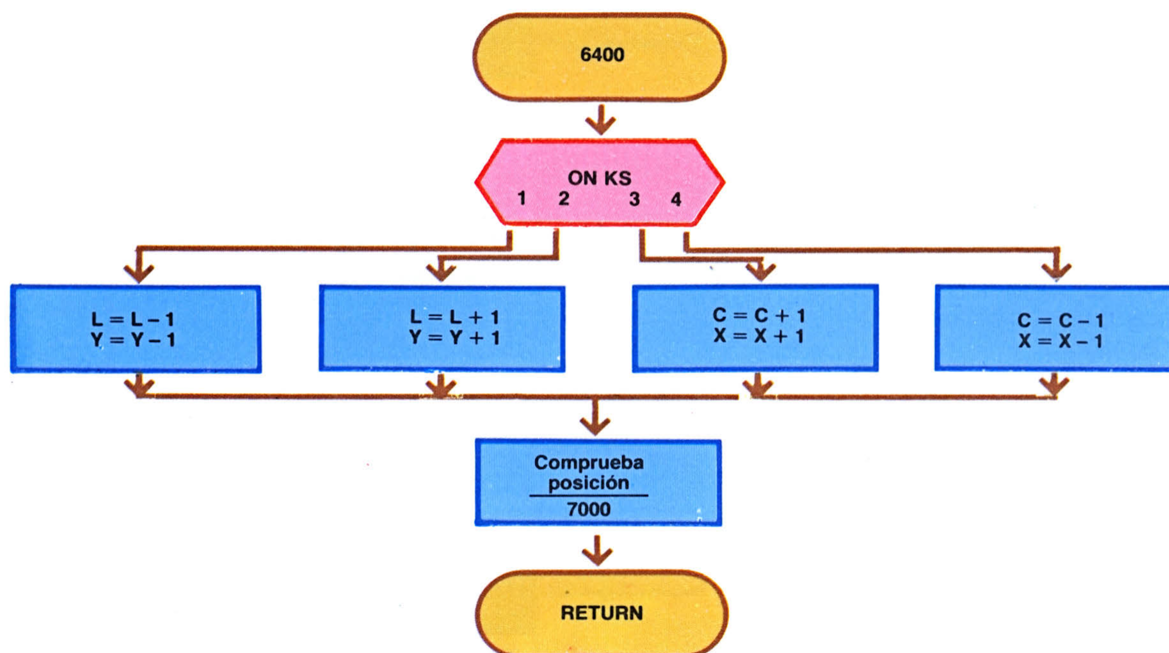
COMPROBACION DEL POSICIONADO

En llamada:
L,C = posición actual correspondiente al buffer
X,Y = posición actual correspondiente a la pantalla
X0,Y0 = coordenadas de principio pantalla útil



DESPLAZAMIENTO DEL CURSOR

En llamada:
KS = contiene el puntero al tipo de desplazamiento deseado, o sea la posición en la que se ha encontrado el código de la tecla en la matriz TS(*)



Obsérvese que las comprobaciones sobre L y C incluyen el caso ">" para que la subrutina pueda utilizarse también para la gestión de las teclas de desplazamiento: efectivamente, para esta última función hay que incrementar la línea o la columna (en base al desplazamiento que se haya solicitado), y en este caso el valor de salida puede ser superior al máximo que se ha previsto.

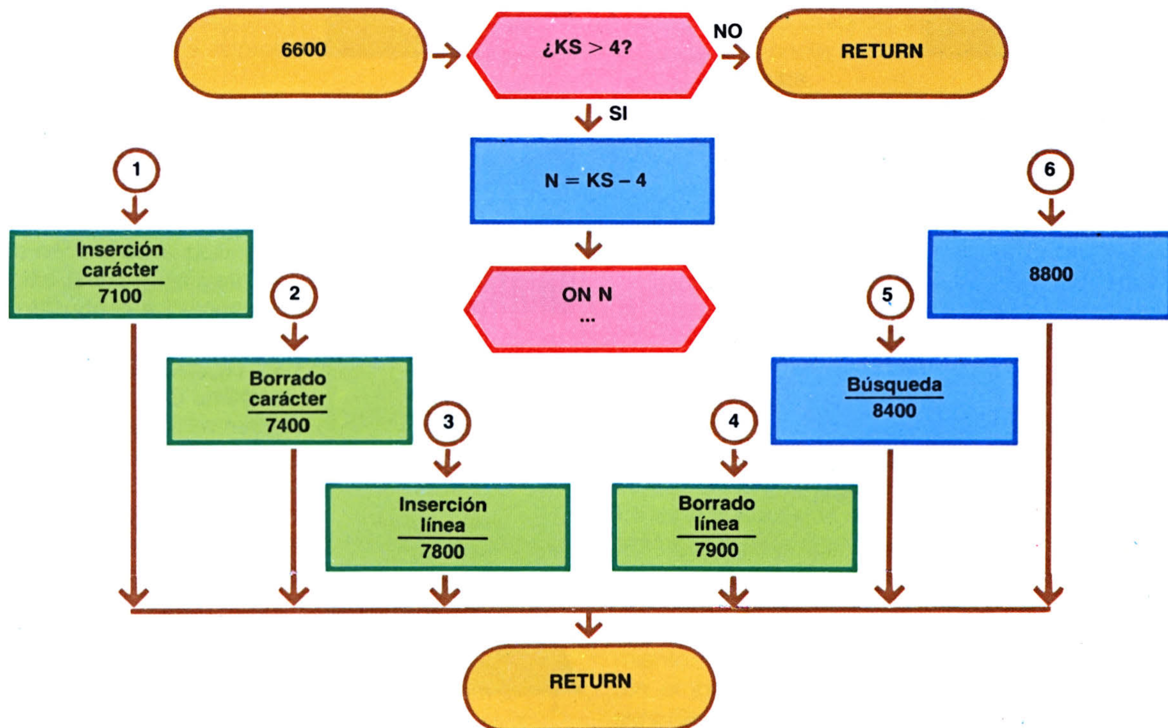
Gestión comandos. En la página siguiente se ha representado el diagrama de principio de la subrutina de gestión comandos. La función es simplemente un test sobre el valor del indicador KS, con la consiguiente llamada a la subrutina de desarrollo.
En la data entry, el flag KS se implanta a un valor correlacionado con una DATA a la función prevista, por lo que la subrutina no necesita control.

Obsérvese que los valores de KS de 1 a 4 son gestionados en otro punto del programa y, por tanto, el primer nivel de selección genera un RETURN para valores inferiores a 4 de KS.
Como en el proceso de la subrutina se entra con KS a partir del valor 5, para utilizar de manera sencilla la instrucción ON KS GOSUB..., hay que indicar antes el contenido de la variable a valores que empiezan por 1. Esta traslación se obtiene restando el valor 4 de KS y utilizando esta nueva variable (N) en la instrucción de salto.

Inserción de un carácter. El diagrama de la página siguiente muestra con detalle el desarrollo de la función de inserción de un carácter.
La lógica seguida consiste en activar el flag KX para que la función permanezca inactiva hasta un comando sucesivo, por lo que KX tiene funciones de «interruptor» que, una vez activado, genera un bucle sobre la rutina 3000 y permite, sin otros comandos, la introducción de una serie de caracteres. Para desactivar el flag, y por tanto terminar la fase de inserción, hay que pulsar la misma tecla empleada para su activación. Por este motivo (funcionamiento como «interruptor»), el flag KX se ha indicado como de tipo ON/OFF en la subrutina 6000.
La funcionalidad descrita implica una repetición de algunas líneas de programa; efectivamente, en la 3000 hay que insertar parte de la lógica utilizada en la 3400 (transferencia al buffer).
Como alternativa se podría parametrizar la 3400, reduciendo así el número de instrucciones duplicadas, aunque generando una lógica menos sencilla de seguir. Además, hay que tener en cuenta que en la 3000 se utiliza la línea reservada al coloquio.
El número de líneas útiles en la pantalla es superior a las 20 previstas para la introducción. Esta diferencia se debe a la necesidad de conservar espacio disponible para los coloquios con el usuario y en esta subrutina se ha mostrado un primer ejemplo. Utilizando una función (en el caso específico de la inserción de un carácter) con lógica de tipo ON/OFF, si no aparece un mensaje que ponga de manifiesto la activación, el usuario no puede recordar si la función aún es operativa o se ha «apagado». Por este motivo se han previsto algunas líneas en la parte inferior de la pantalla en la

GESTION COMANDOS

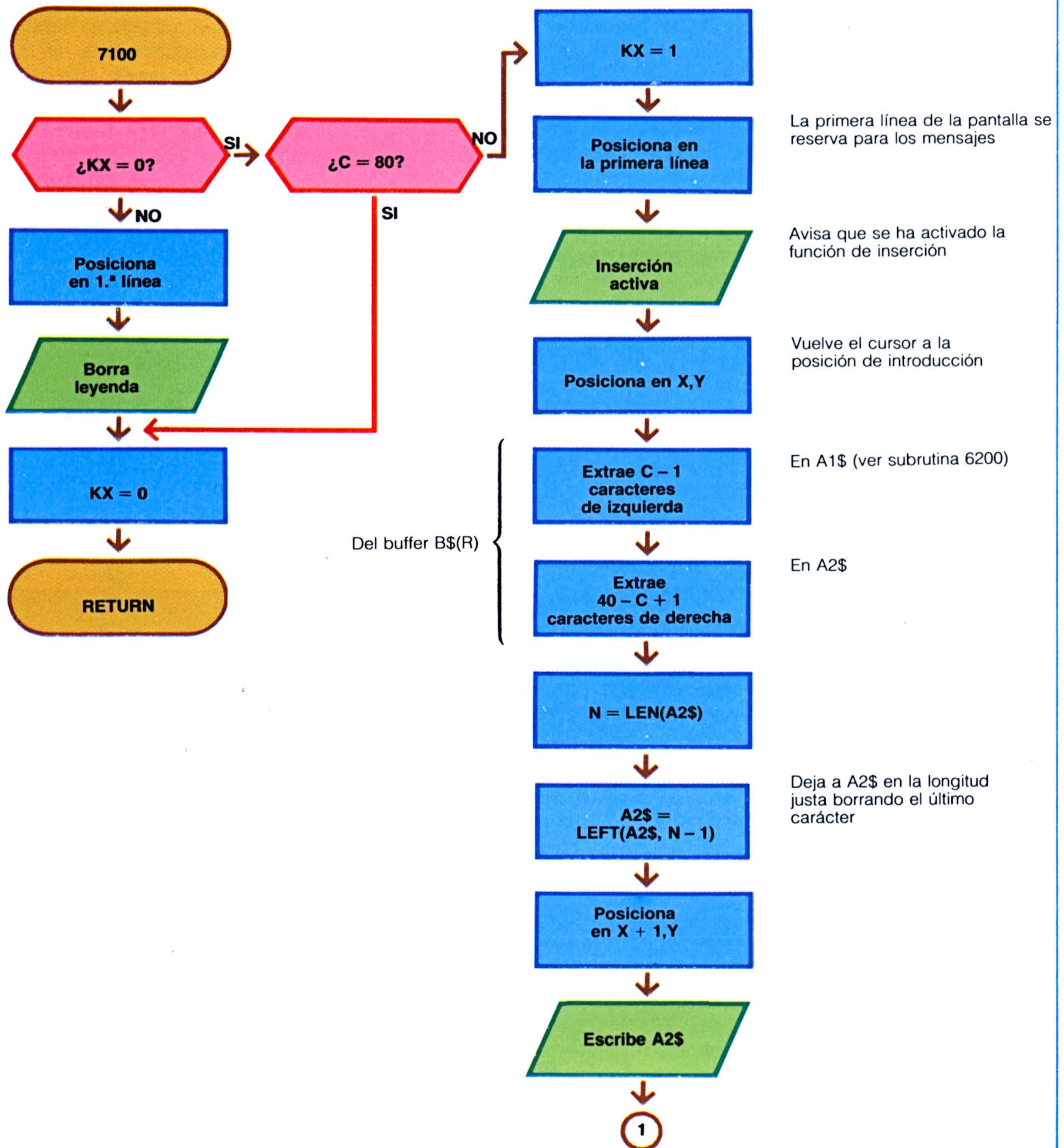
En llamada:
KS = indica el comando pedido. Su valor se determina con un bucle en la subrutina 6000

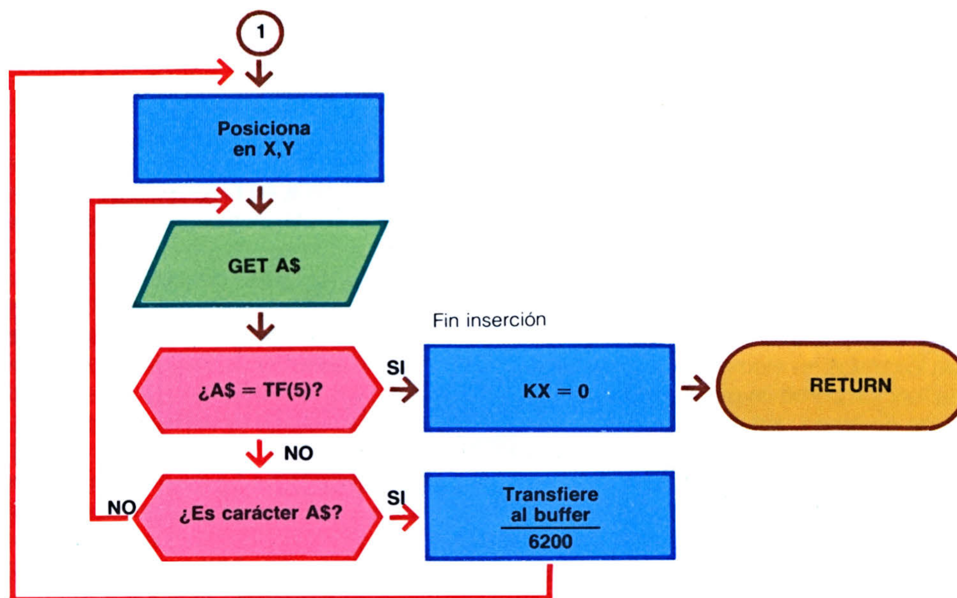


La subrutina 8800, correspondiente a la tecla funcional n. 10, no se activa y permanece disponible para implantaciones

INSERCION DE UN CARACTER

En llamada:
L,C = punteros al buffer y al carácter
X,Y = coordenadas del cursor





que aparecen las distintas leyendas. Estas líneas también se utilizan en otras subrutinas para la introducción de algunos parámetros.
En lo referente a la función opuesta, borrado de un carácter, la subrutina se desarrolla con una lógica análoga. Naturalmente, no se activa el flag ON/OFF y, por tanto, no se emite ningún mensaje. El diagrama correspondiente se ha representado en la página siguiente.

Introducción y borrado de una línea. Las funciones de introducción siguen una misma lógica de base: el borrado, o sea la eliminación de la línea L en la que está posicionado el cursor, y la inserción de una línea vacía encima de la posición del cursor, se realizan en el buffer B1$(*) y al final se vuelve a realizar un ciclo completo de la subrutina de escritura.
Es decir, la variación se realiza primero en el buffer total (B1$(*)) y después se presenta en la pantalla (buffer B$(*)) como una gestión normal.
Como las funciones se realizan con un desplazamiento en el interior del buffer B1$(*), la rutina que sigue a los dos comandos es única, por lo que en la 3000 se implanta el flag KK que será utilizado en la 3400.
Esta última subrutina, cuyo diagrama se muestra en la página 2551, utiliza el flag KK para activar la función requerida de la siguiente manera:

- Si el flag vale 1 (inserción), el contenido de B1$(*) se traslada hacia abajo para dejar una línea vacía en la posición indicada por el cursor
- En el caso opuesto (KK = 2, borrado), el contenido de B1$(*) se traslada hacia arriba para recubrir la línea a borrar con el contenido de la siguiente.

En cada caso, la subrutina prevé el desplazamiento hasta la última línea prevista (dimensión de B1$(*), memorizada en la variable MX). Esto puede comportar pérdidas de tiempo si el buffer no está lleno. Para eliminar este inconveniente puede memorizarse el número más alto de línea escrita y emplear a continuación este valor como límite del bucle. Por motivos de sencillez exclusivamente, en esta versión se ha conservado la lógica descrita.

Búsqueda de una palabra. La subrutina de búsqueda de una palabra en el texto lleva a cabo las siguientes funciones:

- Introducción en W$ (este nombre de variable sólo se considera como ejemplo, por lo que no se encuentra en los listados) de la palabra a buscar
- Búsqueda, en el texto que hay en pantalla, comprobando la coincidencia entre el contenido de B$(*) y W$

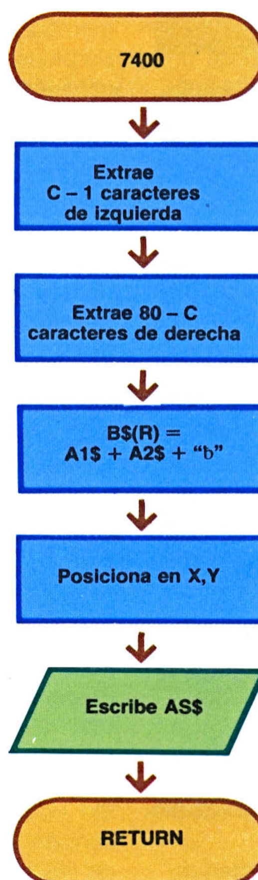
En caso de solución positiva, el cursor se detiene al principio de la palabra (del texto) y, en caso contrario, se posiciona en el primer carácter de la primera línea. En ambos casos, la subrutina vuelve al bloque de introducción para ulteriores comandos.
En esta versión, la búsqueda se explica sólo en el ámbito del contenido de la pantalla; para explorar todo el texto es necesario que el usuario active el desplazamiento.
Sin embargo, la activación de esta función es muy sencilla, puesto que basta con modificar el contenido de NS en caso de solución negativa. De esta manera se obtendrá la presentación de las líneas sucesivas (o precedentes).

Subrutina de impresión. En la versión disco, como se verá más adelante, hay implantada la fusión del fichero texto, creado con el programa de Tratamiento de textos, con un fichero datos (generado por la Base de Datos).
Esta implantación ocupa un importante espacio de memoria, por lo que las funciones de impresión están agrupadas en un programa específico, cargado por el menú principal. Y viceversa, en la versión ilustrada en las páginas 2556 a 2558, para la cual no se activa la fusión, la impresión es una subrutina.
El diagrama representado para la impresión con fusión es idéntico a ésta, excepto un bloque inicial necesario para la carga de los datos en memoria. Efectivamente, estructurando la función como un programa separado, en el momento de su carga, los datos que hay en la memoria se pierden, por lo que al principio del programa hay que insertar una subrutina de lectura del disco análoga a la 2000 usada en el programa de escritura.
Además, en el diagrama no se ha representado el coloquio inicial que permite introducir algunos de los parámetros que se han utilizado en la impresión; los previstos son:

- Número de columnas en impresión. La longitud máxima de una línea es de 80 caracteres (valor asumido por omisión por el programa). Con este flag puede variarse a 40 columnas; la justificación se efectúa con cada petición de impresión para adecuar el texto al número de columnas pevisto
- Número de líneas previstas para cada página, normalmente 20. En el programa de impresión se ha previsto un contador que, al llegar a este valor, produce el salto de página y, si el flag correspondiente es activo, la impresión de la numeración
- Flag que activa la impresión de la numeración de las páginas
- Mensaje de espera, con la correspondiente confirmación por parte del operador de inicio de impresión. Esta operación permite poner en línea la

BORRADO DE UN CARACTER

En llamada:
R,C = punteros al buffer y al carácter a borrar
X,Y = coordenadas del cursor



impresora o modificar la posición del papel antes de empezar la impresión. En esta fase es posible variar el tipo de impresión mediante los códigos de control de la impresora empleada: bastará insertar los códigos necesarios en las líneas de programa utilizadas para inicializar el periférico.

Justificación. La función de justificación se utiliza para obtener el final de cada línea alineado con los otros.

En la fase de escritura de un documento se utiliza siempre el mismo espaciado entre las distintas palabras que componen una línea y, debido a su diferente longitud, el espacio no utilizado al final de la línea es distinto en cada una.

Esta diferencia produce unas irregularidades en el margen derecho que le confieren un aspecto antiestético.

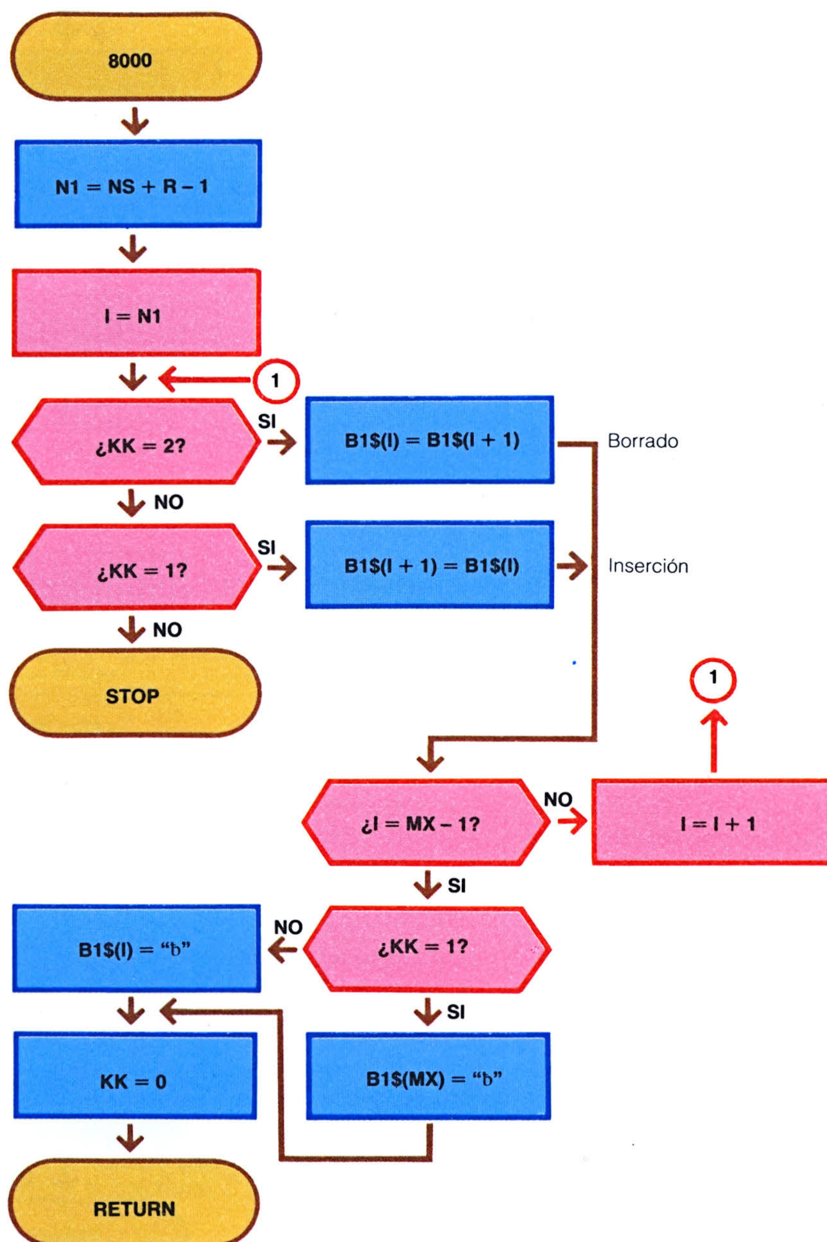
Para eliminar este inconveniente es necesario repartir el espacio no utilizado entre las distintas palabras que constituyen la línea. De esta manera se tiene una distancia distinta entre una y otra palabra, pero todas las líneas terminan en el mismo punto.

La subrutina de justificación realiza la función descrita con la siguiente lógica:

1 / Inicialmente se busca la primera posición no deseada, que indica el final de la línea

2 / La diferencia entre la longitud prevista (80/40 caracteres) y la posición así determinada proporciona el número de espacios a distribuir entre las palabras que constituyen la línea. Naturalmente, este método ofrece un resultado válido sólo a condición de que todas las líneas estén llenas de manera similar

3 / Determinado el número de espacios a ubicar, se llama la subrutina 4200, que modifica la línea insertando un

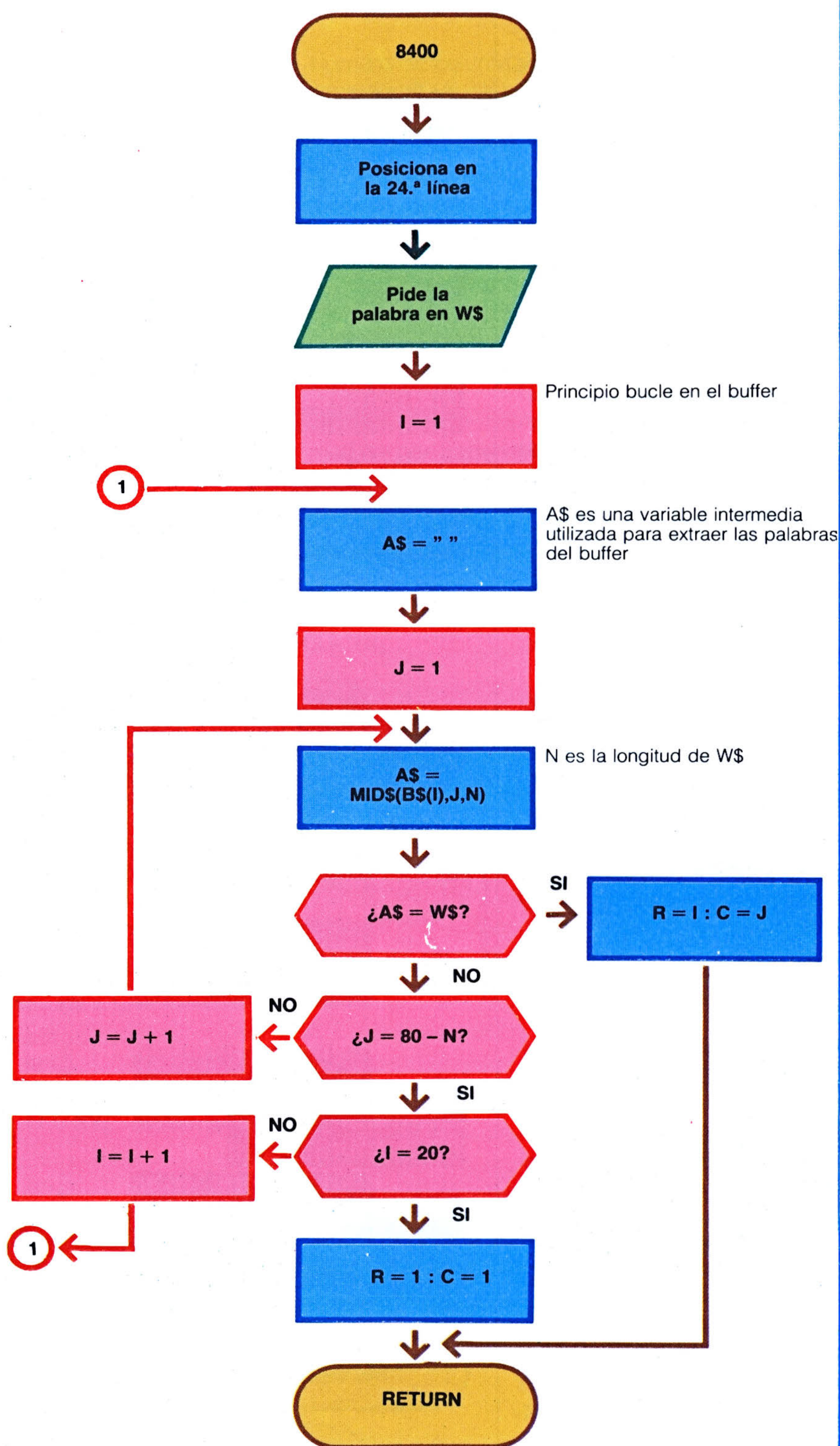


espacio entre cada par de palabras. Esta inserción termina al terminarse los espacios a ubicar, o sea cuando la línea alcanza la longitud prevista.

4 / Las funciones ilustradas se repiten, con un bucle, para todas las líneas previstas

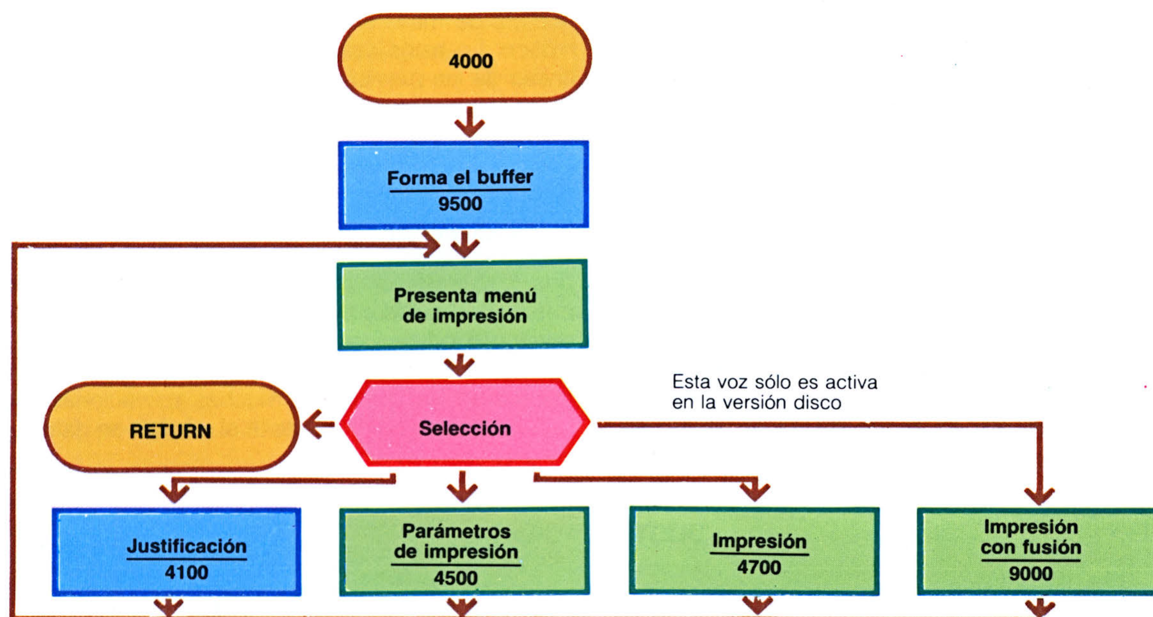
El caso en que la línea termina con el símbolo ">" constituye una excepción de la lógica expuesta. En este caso, el sistema no realiza la justificación porque presupone que el usuario desea regresar al principio.

Justificación de una línea. La subrutina anterior (justificación) extrae cada vez una línea del texto, determina el número de espacios a distribuir y llama, para esta función, la subrutina de justificación de una línea.
La primera función que realiza esta última subrutina es la extracción del buffer actual (o sea de la línea) de la parte escrita (A2$). Siguen el cálculo del número de espacios a ubicar y el bucle de ubicación. El bucle consiste en una extracción, partiendo del final, de cada carácter (y por tanto de la cadena A2$, cuyo nombre sólo es indicativo) y en la comprobación de que este carácter sea adyacente a un espacio. En caso de solución positiva, el carácter extraído constituye el principio de una palabra y, por tanto, determina la posición de inserción de uno de los espacios a ubicar.
Para realizar este bucle es necesario utilizar un indicador de tipo ON/OFF. Efectivamente, si el carácter extraído es un espacio, hay que comprobar si va precedido de otro espacio. En este caso no identifica una palabra, sino un doble espaciado entre dos palabras y, por tanto, no debe incrementar el contador. Es decir, el bucle anterior sirve para contar las pa-



La subrutina es en versión generalizada para ser aplicada al C-64 y al Apple. No aprovecha plenamente el Basic MSX, que podría ejecutar la misma función con una sola instrucción

SUBRUTINA DE IMPRESION



Los parámetros de impresión previstos en esta versión son:
PC$ = código de tipo impresión (compactada, expandida, etc.). El valor depende de la impresora utilizada (*)
N1 = columna principio impresión (N1 = 1) (*)
NN = número de columnas (NN = 80)
RR = líneas por página (LL = 40)
FN = flag de numeración (FN = 0, OFF)

(*) No activos en esta versión

El detalle de la 4500 (introducción parámetros impresión) no se muestra, porque se reduce a un simple coloquio

La impresión se presenta como si fuese una subrutina, mientras que en la versión disco indicada, es un programa propiamente dicho. Naturalmente, debe estar incluida la lectura del fichero datos (en B1$(*))

labras que componen una línea. Esta función se realiza contando los espacios de separación entre una palabra y otra, pero como dos espacios sucesivos no deben considerarse como dos separaciones, se justifica la necesidad de un flag que indique, al encontrar un espacio, si va precedido de un carácter o de otro espacio.
En el primer caso puede incrementarse el contador porque ya se ha conseguido la identificación del byte de final de una palabra (o mejor, de principio, puesto que el bucle procede de derecha a izquierda); en el segundo se trata de un doble espacio y el contador no debe incrementarse.
Al final del bucle, la variable M contiene el número de «separaciones» halladas y la matriz P(40), la columna en la que hay cada separación.
El número total de espacios a ubicar está en la variable N1: por tanto, el número de espacios a añadir en cada separación viene dado por el cociente N1/M, mientras que la posición en la que deben ubicarse estos espacios se indica en la matriz P(40). Por tanto, sólo queda por activar un bucle que sume un número de espacios igual a N1/M en cada una de las posiciones indicadas en P(*).

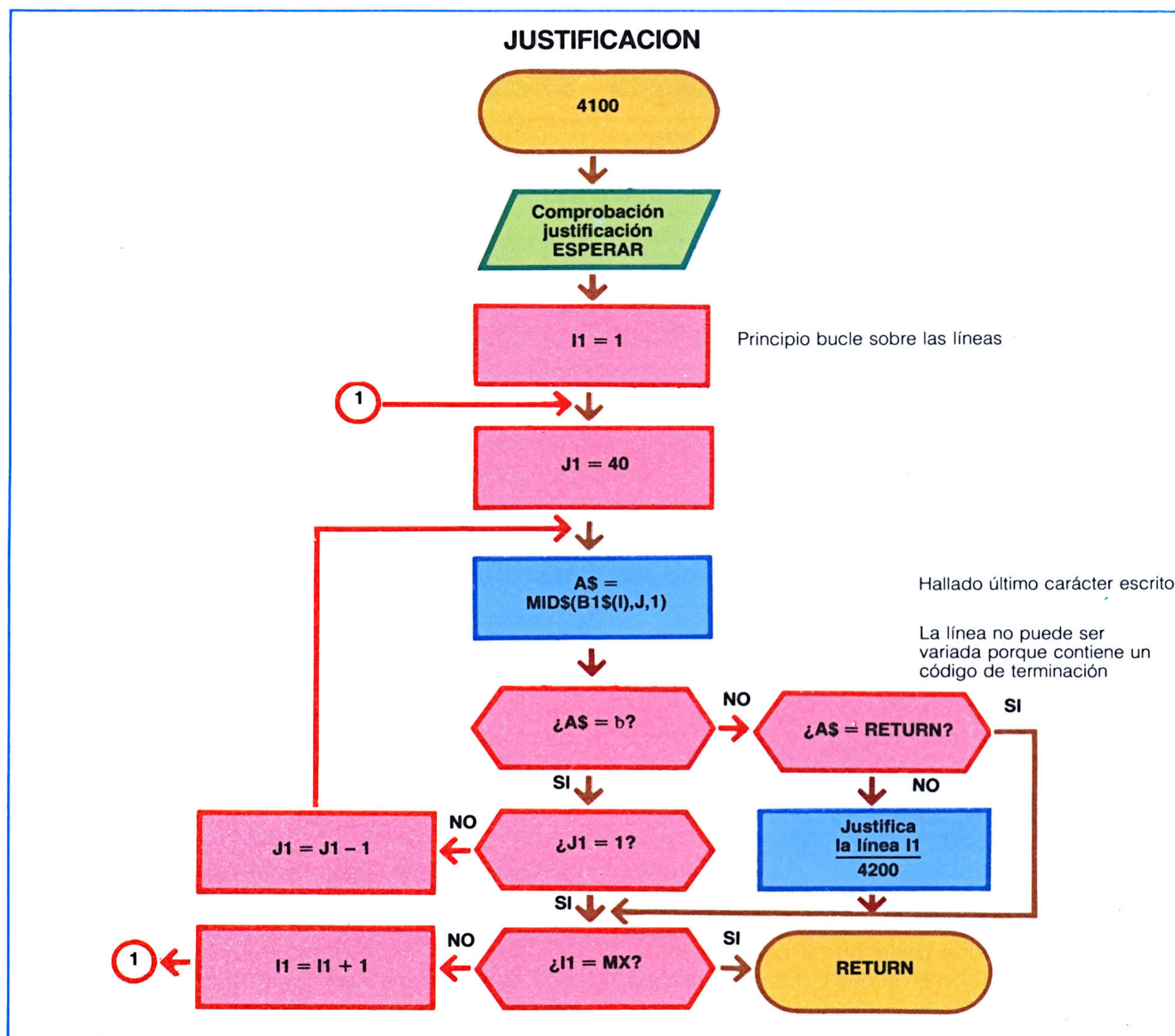
Impresión. La subrutina de impresión debe prever dos funciones particulares:

- Emisión del código de control de la impresora
- Formación de la línea de 80 caracteres

La primera función puede emplearse para implantar un determinado tipo de impresión. Naturalmente, para activarla hay que conocer los códigos de la impresora utilizada; además, para muchos tipos de impresión, no basta la emisión de un solo código, puesto que un comando se obtiene con una secuencia. En el diagrama se ha indicado la función a título de ejemplo más que para su empleo real, quedando para el usuario las necesarias implantaciones.
La parte «útil» de la rutina está constituida por un bucle que extrae una línea del buffer B1$(*), examina su contenido carácter por carácter y, en base al resultado del texto, imprime o no.

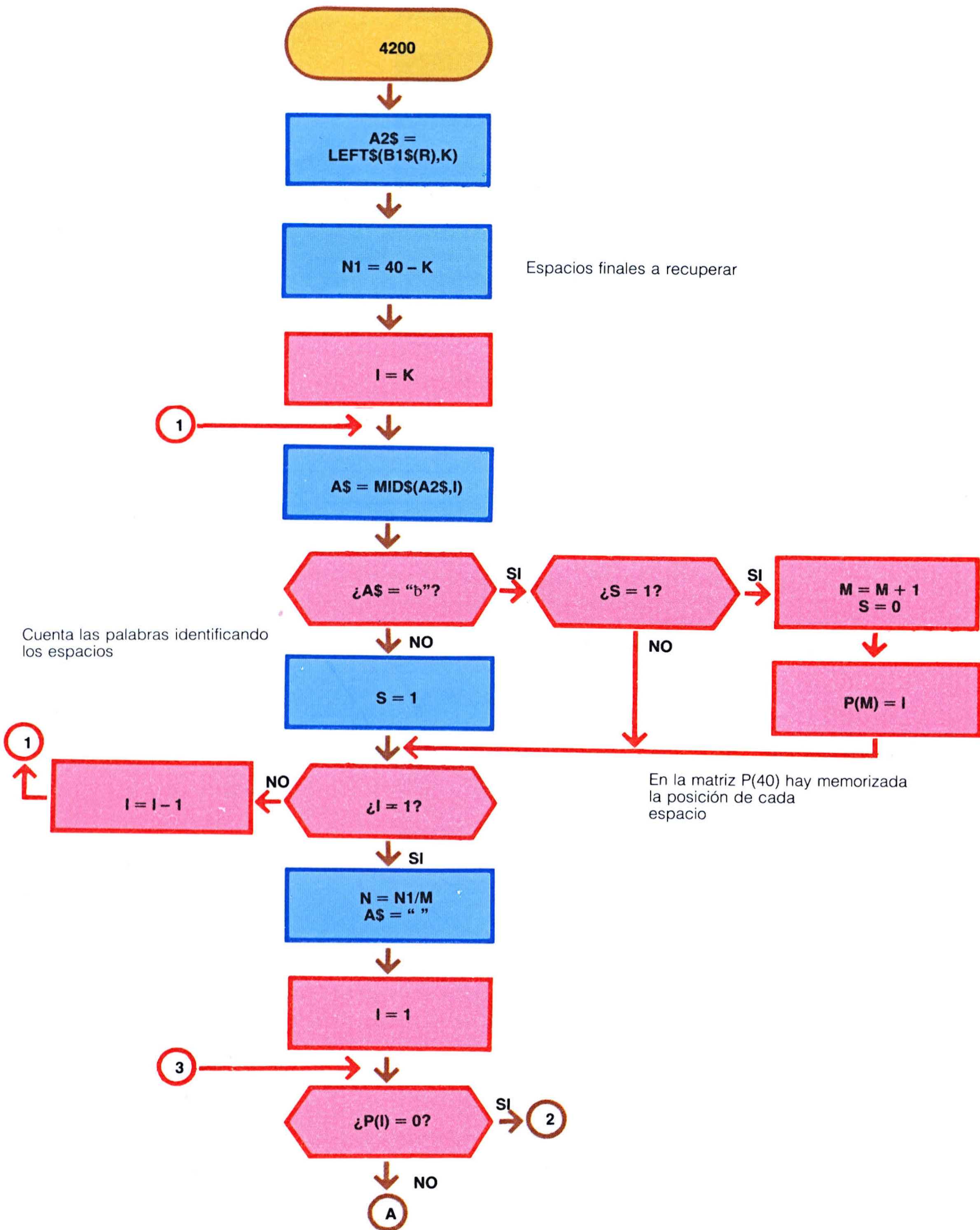
Esta lógica particular es consecuencia de la necesidad de sumar dos líneas de vídeo para obtener una línea de impresión.
La función, de por sí bastante sencilla, comporta una complicación en la fase de impresión. Efectivamente, al sumar dos líneas de vídeo para obtener una de impresión, puede suceder que se unan en una sola dos líneas que estén separadas en realidad por un retorno al principio. Para la formación del buffer B1$(*), esta eventualidad no constituye ninguna complicación, mientras que en la impresión debe tenerse en cuenta, porque hay que ordenar el retorno del carro y el salto de línea.
Como se ha indicado en la ilustración de la data entry para el Tratamiento de textos, el buffer debe contener también los eventuales códigos RETURN, que indican una vuelta al principio. Por tanto, en la fase de impresión hay que analizar el contenido de la línea carácter por carácter y, al encontrar el código RETURN, activar la función CR + CF (retorno de carro y salto de línea). Además, hay que tener en cuenta que, en muchos casos, la impresora tiene un control hardware sobre el número de líneas que ordena un salto de página automático. En estos casos, al imprimir un texto con más líneas de las previstas por el salto de página se tiene su interrupción de acuerdo con cada final de página automático; por tanto, hay que excluir la función.
Esta exclusión podrá obtenerse, según la impresora, bien vía hardware, implantando determinados interruptores en el interior del periférico, bien software, enviando un código de comparación particular. En ambos casos se trata de funciones estrechamente ligadas a la máquina y para las cuales no es posible proporcionar indicaciones generalizadas.

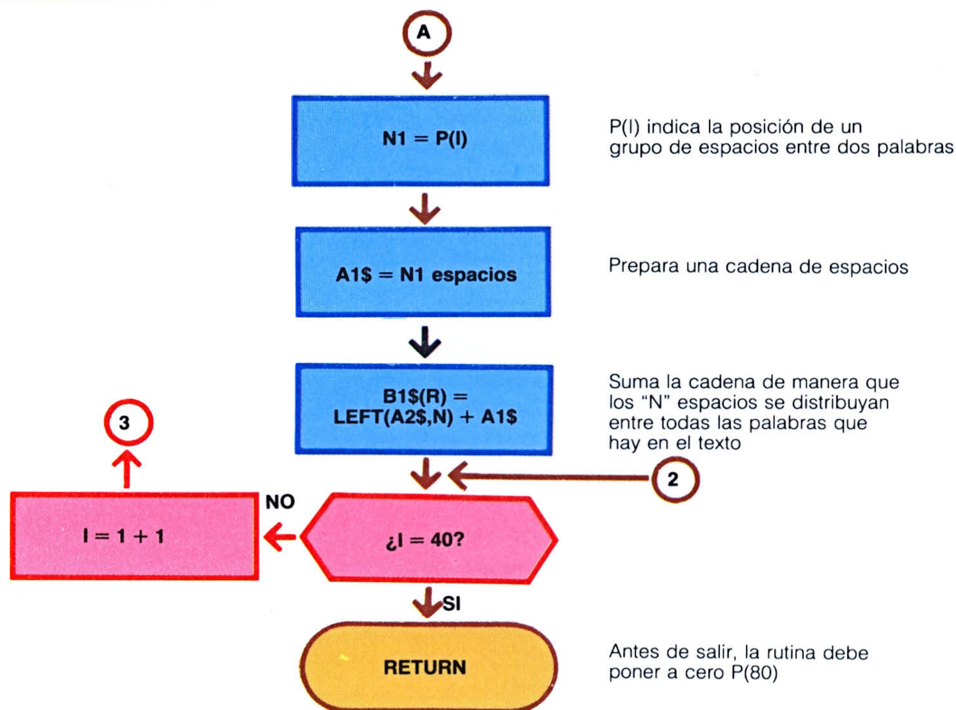
Impresión con fusión. En muchas aplicaciones, la impresión final también necesita el empleo de datos que no aparecen en el texto.

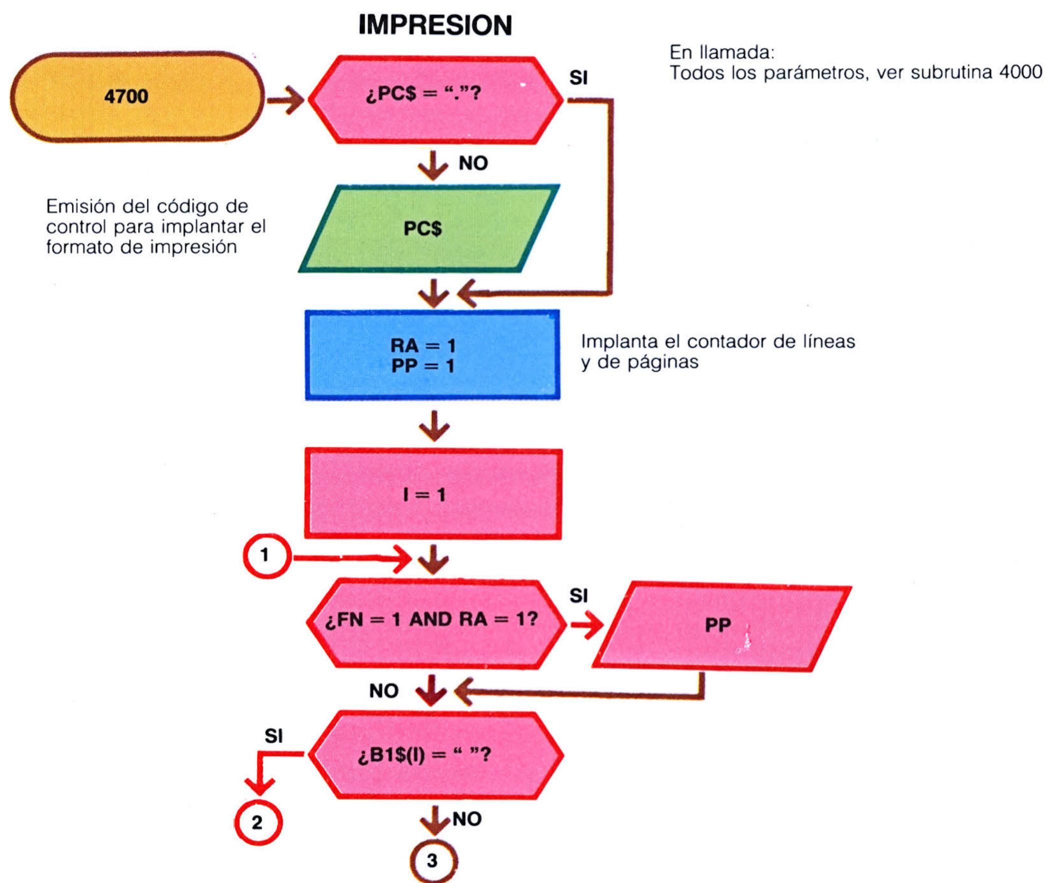


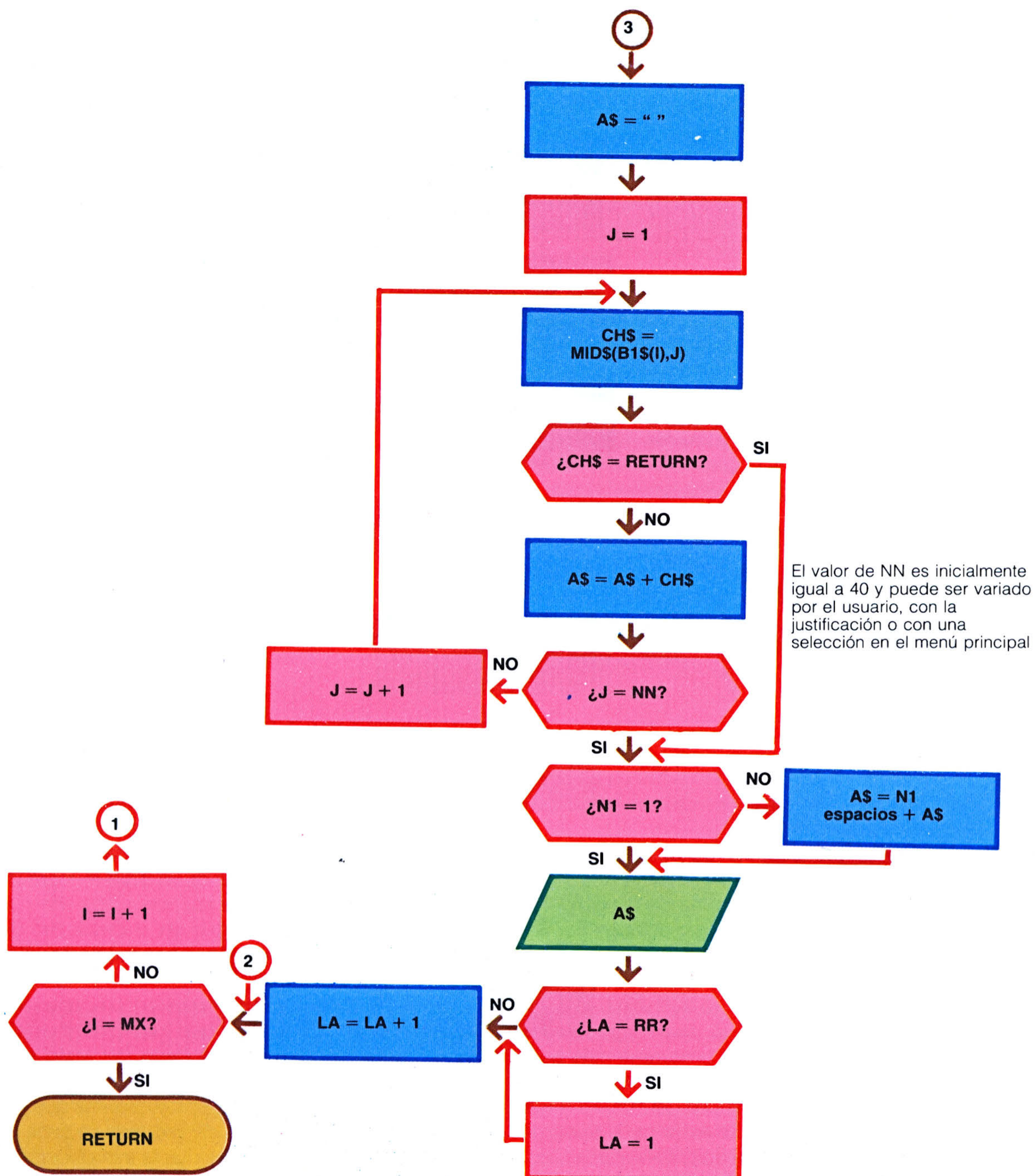
JUSTIFICACION DE UNA LINEA

En llamada:
L = línea actual (L = I1 en el bucle de la 4100)
K = última posición escrita (K = J1 en el bucle sobre los caracteres de la 4100)

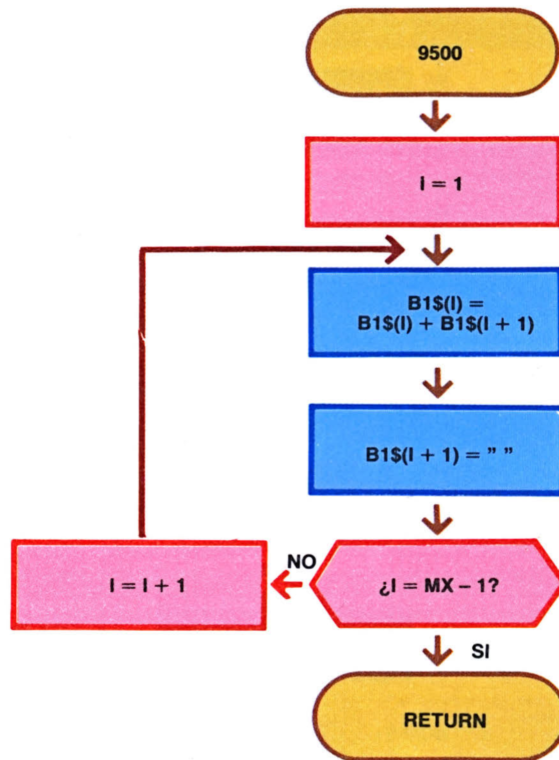



A
N1 = P(I)
P(I) indica la posición de un grupo de espacios entre dos palabras
A1$ = N1 espacios
Prepara una cadena de espacios
B1$(R) = LEFT(A2$,N) + A1$
Suma la cadena de manera que los "N" espacios se distribuyan entre todas las palabras que hay en el texto
3
2
NO
I = 1 + 1
¿I = 40?
SI
RETURN
Antes de salir, la rutina debe poner a cero P(80)


IMPRESION
4700
¿PC$ = "."?
SI
En llamada:
Todos los parámetros, ver subrutina 4000
NO
Emisión del código de control para implantar el formato de impresión
PC$
RA = 1
PP = 1
Implanta el contador de líneas y de páginas
I = 1
1
¿FN = 1 AND RA = 1?
SI
PP
NO
SI
¿B1$(I) = " "?
2
NO
3

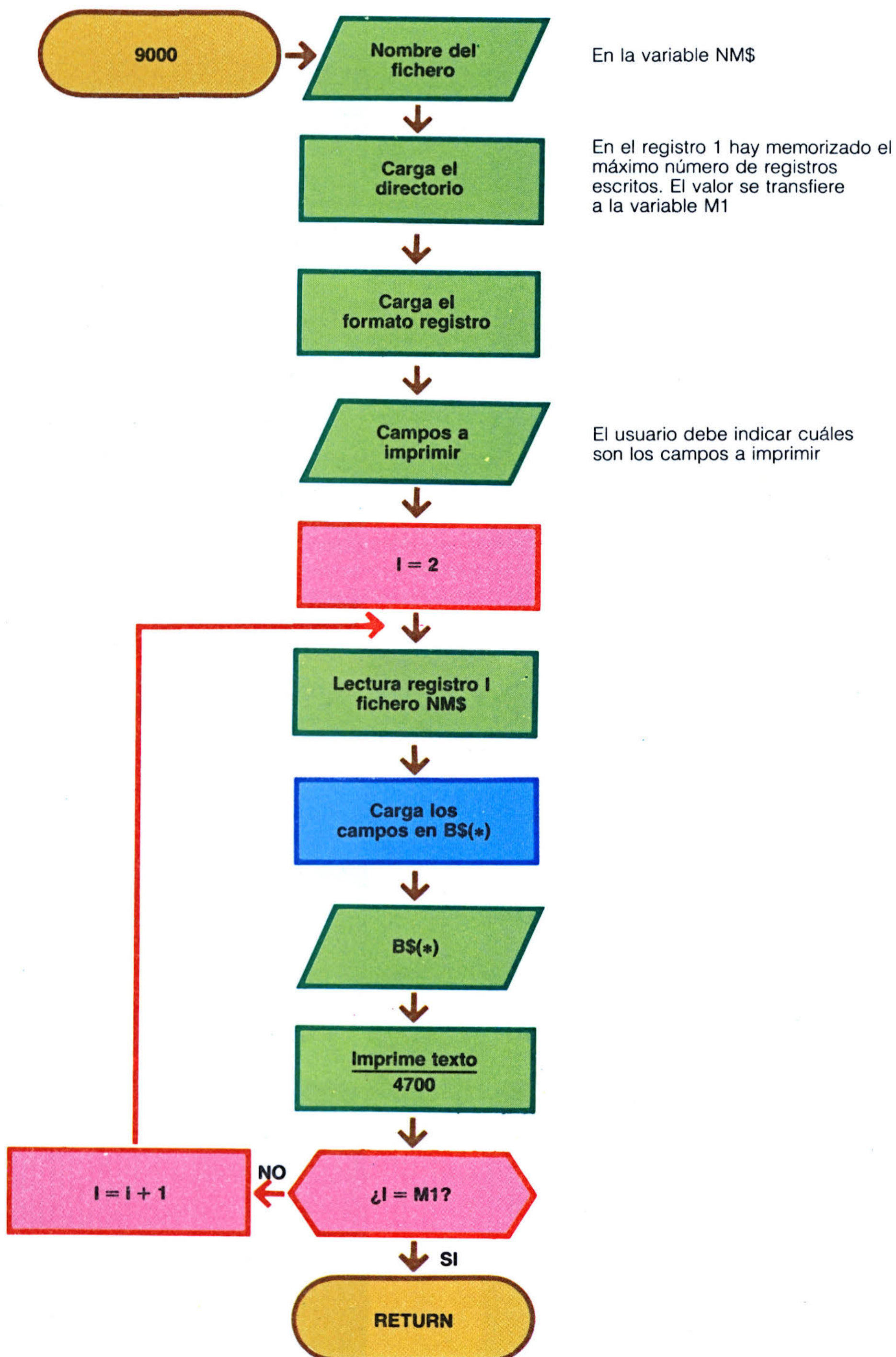


El valor de NN es inicialmente igual a 40 y puede ser variado por el usuario, con la justificación o con una selección en el menú principal



Por ejemplo, en la preparación de cartas es muy frecuente el caso en el que el texto no varía, mientras que hay que modificar la cabecera.
En estos casos es útil disponer de una función que permita unir datos procedentes de un fichero con los que pertenecen al texto; el diagrama de la página siguiente muestra los distintos puntos a desarrollar para obtener una lógica de este tipo.
Como el empleo principal de esta función es la unión de direcciones en un solo texto, la subrutina está estructurada de manera que utilice datos introducidos con la Base de Datos, que es el programa más adecuado para gestionar un listín de direcciones. Activando esta subrutina se pide el nombre del fichero datos del que se tomarán las direcciones (u otros tipos de datos, según el significado en la aplicación particular). Siguen las funciones de carga del formato de registro, realizadas de manera análoga a las previstas en la Base de Datos, y la selección de los campos a imprimir. La salida se obtiene con un bucle de lectura de cada registro del fichero datos, impresión de su contenido (o de la parte seleccionada) e impresión del texto contenido en la memoria. Como se ha indicado, la aplicación más característica de esta función es la confección de cartas circulares: el texto de la carta se prepara con el programa de Tratamiento de textos, mientras que la cabecera (dirección) puede tomarse de un fichero de direcciones, de una nómina, etc. Esta solución evita tener que preparar un gran número de textos diferentes sólo porque las cabeceras son distintas y muestra la lógica a seguir en el desarrollo de una determinada implantación, generalmente llamada «función de textos».
En las activaciones que requieren preparaciones de textos con una frecuencia que justifique el empleo de una máquina, normalmente parte del texto es recurrente. Por ejemplo, las cartas comerciales empiezan y terminan siempre con frases protocolarias que pueden repetirse; en la redacción de informes se tiene una situación análoga, etc. Para evitar la introducción de las partes repetitivas en cada documento, puede adoptarse un método análogo al anterior, o sea memorizar la parte recurrente en un fichero y unirla al texto en la fase de impresión. Naturalmente, hay que hacer algunas modificaciones, como por ejemplo la eliminación de la elección de los campos. Efectivamente, en este caso, el fichero no es generado por la Base de Datos y el registro no está subdividido en campos.

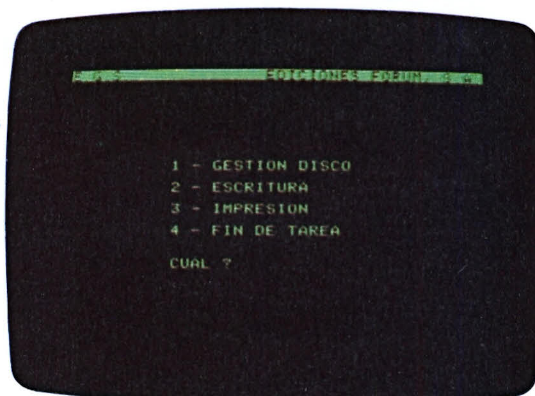
IMPRESION CON FUSION



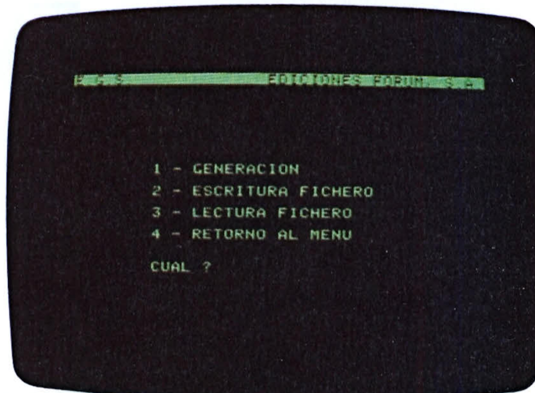
Para el detalle de las funciones de carga de los ficheros NM$ véase el programa Base de Datos

La subrutina no está prevista en la versión cinta

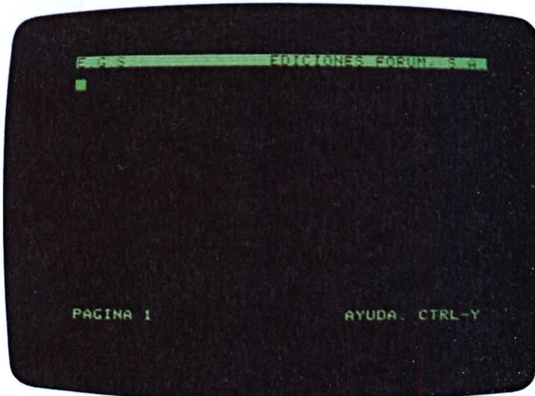
Tratamiento de textos. El programa en la versión DOS tiene un menú principal en el que se pide la función deseada. Por ejemplo, tecleando el carácter 3 se activa la carga del programa de impresión.



Gestión disco. Se ha activado la voz 1 del menú principal: el menú secundario presentado en esta foto permite realizar las operaciones indicadas. La Generación es esencial para la escritura de un nuevo texto (con esta operación se implantan los parámetros de escritura).



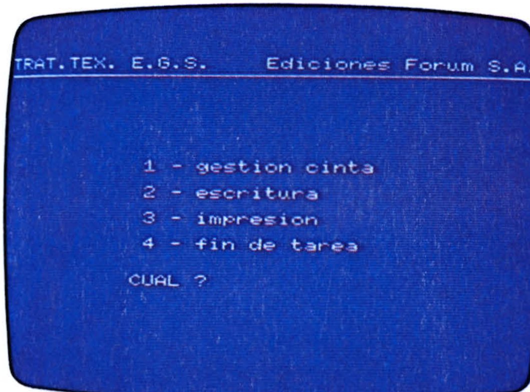
Escritura. La fase de escritura permite crear un texto o modificar uno ya existente en el disco datos; en este caso, antes de pasar a la escritura, hay que haber cargado el fichero datos (voz 3 del menú anterior). En escritura, los comandos disponibles son:
CTRL + I = inserción carácter
CTRL + D = borrado carácter
CTRL + P = página adelante
CTRL + O = página atrás
CTRL + L = inserción línea
CTRL + K = borrado línea
CTRL + R = búsqueda palabra
CTRL + E = introducción y retorno al menú.



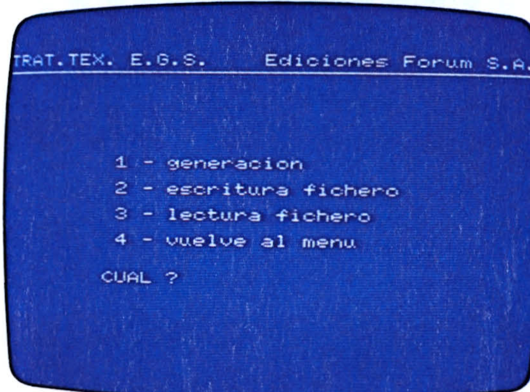
Impresión. Para el programa de impresión, en primer lugar hay que leer los datos (o sea el texto a imprimir) del disco que los contiene; para ello, el usuario debe introducir el nombre del fichero en el que hay el texto. A continuación se presentará un menú secundario que contiene las siguientes funciones:
– Justificación
– Definición de los parámetros de impresión
– Impresión
– Impresión con fusión.



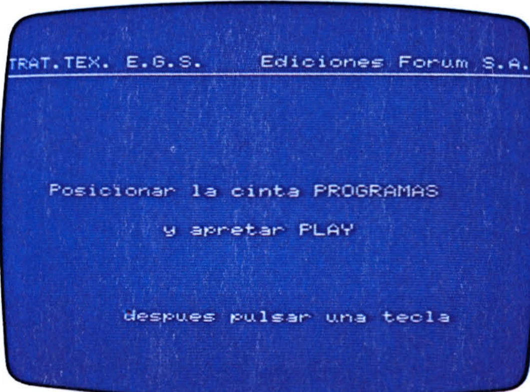
Tratamiento de textos. También la versión MSX, que adopta como soporte la cinta magnética, prevé un menú principal. La voz 2 predispone la pantalla vídeo para la escritura del texto, durante la cual se dispone de los siguientes comandos:
F3 = inserción carácter
F4 = borrado carácter
F5 = inserción línea
F6 = borrado línea
F7 = página atrás
F8 = página adelante
F9 = búsqueda palabra
F1 = ayuda
F2 = introducción y retorno al menú.



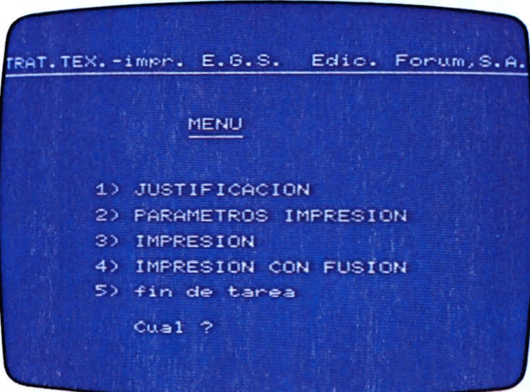
Gestión cinta. El menú secundario mostrado en la foto indica las funciones útiles para crear, grabar o leer el fichero datos en la (o de la) cinta. La Generación (voz 1) se utiliza únicamente para implantar los parámetros con el número de páginas y el número de líneas por página.



Impresión. Al seleccionar la voz 3 del menú principal se activa el procedimiento de carga del programa correspondiente. El usuario debe proceder de acuerdo con las instrucciones proporcionadas por la máquina.



Una vez cargado el programa de impresión y tecleado el RUN deberá leerse el fichero datos que contiene el texto a imprimir. Sólo después de esta operación se presentará el menú de esta foto.





Tratamiento de textos/Versión Commodore 64

Tratamiento de textos

```
10 REM ******************************
12 REM *                            *
20 REM *   TRATAMIENTO DE TEXTOS    *
21 REM *        VERSION C-64        *
22 REM *                            *
30 REM ******************************
35 POKE53280,14:POKE53281,6:POKE650,0:PRINTCHR$(142)CHR$(8)
40 PRINT"":PRINT"INTRODUCIR EL NUMERO CORRESPONDIENTE"
42 PRINT"EL PROGRAMA DESEADO"
44 PRINT"<1> GENERACION ARCHIVOS"
46 PRINT"<2> IMPRESION"
47 PRINT"<3> FIN DE TAREA"
50 INPUT"ELECCION : ";R
55 IF R<1 OR R>3 THEN 40
56 IF R=3 THEN PRINT"":END
57 PRINT""SPC(4)"ESPERAR LA CARGA, POR FAVOR"
60 ON R GOTO 100,200
100 PRINT"POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
110 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"GA"CHR$(34)",8"
120 GOTO1000
200 PRINT"POKE43,1:POKE44,8:POKE45,3:POKE46,8:POKE47,3:POKE48,8"
210 PRINT"POKE49,3:POKE50,8:POKE4096,0:LOAD"CHR$(34)"SP"CHR$(34)",8"
1000 POKE198,10:FORL=631TO640:READN:POKEL,N:NEXT
1010 DATA19,13,145,145,13,82,85,78,58,13
1050 REM :
1051 REM   GA = GENERACION
1052 REM
1053 REM   SP = IMPRESION
```

```
10 REM ************************
15 REM * TRATAMIENTO DE TEXTOS *
20 REM ************************
30 HO$="":POKE 650,128:POKE 53280,14:POKE 53281,6
40 DIM H$(12),B$(20):DIM B1$(402):MX=40:NP=1:PG=20
50 RESTORE:FOR I=1 TO 9:READ H$(I):NEXTI:FOR I=1 TO 8:READ TF(I):NEXT I
55 GOSUB 1300
60 IF OP=3 THEN PRINTHO$:POKE 646,14:END
100 ON OP GOSUB 2000,3000
110 GOTO 50
500 REM * CABECERA *
505 PRINTHO$;CHR$(14);:X=13:Y=2:GOSUB11000:PRINT"";IN$
510 Y=Y+1:GOSUB11000:PRINT"----------------"
515 B$="          ":BA$=B$+B$+B$+B$:B$(0)="-------":B$=""
520 FI$="":FOR I=1 TO 40:FI$=FI$+CHR$(35):NEXT I
535 RETURN
1300 REM
1302 REM MEDIDA:  ----------
1305 PRINTHO$:B$="          ":BA$=B$+B$+B$+B$:B$(0)=BA$
1306 BU$="":FOR J=1 TO 40:BU$=BU$+CHR$(32):NEXT J
1307 PRINTCHR$(14):POKE 53280,14
1310 PRINT"           .I.♥.   DICIONES  ORUM. ♥.♠."
1312 PRINT"   ENU     ","                          "
1314 PRINT"           ","   RAT. TEXTOS - GENERACION     "
1316 PRINT"                                        "
1320 FC=1:NV=3
1325 FOR J=1 TO NV:READ VO$(J):NEXT J
1327 Y=10:X=10
1330 FOR J=1 TO NV:GOSUB11000:PRINT"";J;"-";" ";VO$(J)
1332 Y=Y+2:NEXT J
1335 X=2:Y=20:GOSUB11000:PRINT"CUAL ?":X=10
1337 GET OP$:IF OP$<>"" THEN 1360
1340 GOSUB11000:PRINTCHR$(188)
1342 FOR T=1 TO 30:NEXT T
1344 GOSUB11000:PRINTCHR$(172)
1346 FOR T=1 TO 30:NEXT T
1348 GOSUB11000:PRINTCHR$(187)
1350 FOR T=1 TO 30:NEXT T
1352 GOSUB11000:PRINTCHR$(190)
1354 FOR T=1 TO 30:NEXT T
1356 GOTO 1337
1360 OP=VAL(OP$):IF OP<=0 OR OP>NV THEN 1337
1365 X=10:Y=20:GOSUB11000:POKE199,1:PRINTOP
1370 FOR T=1 TO 1500:NEXT T
1400 RETURN
2000 REM *** GESTION DISCO ***
2004 REM RESTORE SUI DATI
2005 RESTORE:FOR J=1 TO 20:READ X$:NEXT J
2010 NV=4
2015 IN$="IESTION  DISCO":GOSUB 500
2020 GOSUB 1325
2025 IF OP=4 THEN RETURN
2030 IN$="IRAT. DE TEXTO":GOSUB 500:X=1:Y=13:GOSUB11000
2031 PRINT"                    ------"
2035 X=15:Y=13:GOSUB11000:INPUT C$
2037 FL$="":FOR KL=1 TO 6:CK$=MID$(C$,KL,1):IF CK$="-" THEN CK$=CHR$(32)
2038 FL$=FL$+CK$:NEXT KL:C$=FL$
2040 NF$=C$:IF NF$="" THEN 2035
2042 GOSUB 6000:IF OP<>1 THEN 2100
2044 GOSUB 40000
2046 GOSUB 500
2050 X=9:Y=13:GOSUB11000:INPUT"NUMERO PAGINAS ";NP
2052 IF NP<=0 THEN 2050
2055 X=9:Y=15:GOSUB11000:INPUT"LINEAS POR PAGINA ";PG
2057 IF PG<=0 THEN 2050
2060 MX=PG*NP
2065 IFMX>400THENGOSUB500:X=1:Y=23:GOSUB11000:PRINT"MAX 400 LINEAS DE TEXTO":GOT
O2050
2100 IF OP=3 THEN 2170
2105 GOSUB500:A$="CREACION":IF OP=2 THEN A$="GRABACION"
2110 FOR I=1 TO MX
2115 IF OP=1 THEN B1$(I)=FI$
2120 X=1:Y=23:GOSUB11000:PRINT"";A$;" LINEA ";I
2125 GOSUB 45000
2130 AA$=B1$(I):PA=I+1:GOSUB 6800:PRINT#2,AA$
2140 NEXT I
2142 INPUT#1,AD,X$,CD,DD
2145 IF OP<>1 THEN 2170
2147 IF MX<10 THEN A1$="00"+RIGHT$(STR$(MX),1):GOTO 2150
2148 IF MX<100 THEN A1$="0"+RIGHT$(STR$(MX),2):GOTO 2150
2149 A1$=RIGHT$(STR$(MX),3)
2150 IF PG<10 THEN A2$="0"+RIGHT$(STR$(PG),1):GOTO 2155
2152 A2$=RIGHT$(STR$(PG),2)
2155 AA$=A1$+A2$+LEFT$(FI$,35):PA=1:GOSUB 6800:PRINT#2,AA$
2160 GOSUB 6900:GOTO 2005
2170 GOSUB500:GOSUB 40000
2171 PA=1:GOSUB6800:INPUT#2,AA$
2174 MX=VAL(LEFT$(AA$,3))
2180 PG=VAL(MID$(AA$,4,2))
2185 FOR I=1 TO MX
2190 PA=I+1:GOSUB 6800:INPUT#2,AA$
2195 X=1:Y=23:GOSUB11000:PRINT"LECTURA LINEA : ";I
2200 B1$(I)=AA$:GOSUB 46000:NEXT I
2205 GOSUB 6900:GOTO 2005
2500 REM  --- --- --- --- --- --- ---
2505 L=LEN(Z$):IF L=0 THEN RETURN
2510 FOR I=1 TO L:C=ASC(MID$(Z$,I,1)):IF C<>32 THEN I1=I:GOTO 2520
2515 NEXT I:Z$="":RETURN
2520 FOR I=L TO 1STEP-1:C=ASC(MID$(Z$,I,1)):IF C<>32 THEN I2=I:GOTO 2530
2525 NEXT I
2530 K0=I2-I1+1:Z$=MID$(Z$,I1,K0):RETURN
2900 REM
2910 D$=LEFT$(BU$,30):X1=0:Y1=YT:GOSUB11100:PRINTD$;:RETURN
3000 REM *** ESCRITURA ***
3005 PRINTHO$;CHR$(14):POKE 53280,6
3010 XC=0:L0=40:CC=1:FOR I=1 TO 20:B$(I)=B1$(I+NS):X=XC:Y=I+2:GOSUB11000
3012 IF B$(I)="" THEN B$(I)=BA$
3014 PRINTB$(I);:NEXT I:X=25:Y=24:GOSUB11000:PRINT"CTRL+Y : AYUDA";
```

```
3015 X=1:Y=24:GOSUB11000:PRINT"PAGINA ";NS/20+1;:POKE 646,1
3015 IF K<>0 THEN CC=K:K=0
3020 YC=CC+2:GOSUB 3500
3030 IF A0=140 THEN GOSUB 3400:RETURN
3040 IF A0<>17 THEN 3070
3050 CC=CC+1:IF CC>20 THEN 3105
3060 GOTO 3020
3070 IF A0<>145 THEN 3100
3080 CC=CC-1:IF CC<1 THEN 3135
3090 GOTO 3020
3100 IF A0<>133 THEN 3130
3105 GOSUB 3400
3110 NS=NS+20:IF NS>=MX THEN NS=NS-20
3120 GOTO 3010
3130 IF A0<>137 THEN 3160
3135 GOSUB 3400
3140 NS=NS-20:IF NS<0 THEN NS=0
3150 GOTO 3010
3150 IF A0<>135 THEN 3200
3165 GOSUB 3400
3170 FOR I=MX-1 TO NS+CC STEP -1
3180 B1$(I+1)=B1$(I):NEXT I
3190 B1$(NS+CC)=BA$:K=CC:GOTO 3010
3200 IF A0<>139 THEN 3250
3205 GOSUB 3400
3210 FOR I=NS+CC+1 TO MX
3220 B1$(I-1)=B1$(I)
3230 NEXT I:B1$(MX)=BA$
3240 K=CC:GOTO 3010
3250 IF A0<>136 THEN 3020
3255 GOSUB 3400
3260 X1=0:Y1=24:GOSUB11100:PRINTLEFT$(BU$,39);:Y=Y1:X=3:GOSUB11000:PRINT"↑L↑I_↑
?";
3261 X=X+9:R$="":FOR P=1 TO 15
3262 GET F$:IF F$="" THEN 3262
3263 IF F$=CHR$(13) THEN P=15:GOTO 3270
3265 R$=R$+F$:X=X+1:GOSUB11000:PRINTF$;
3270 NEXT P
3272 IF PEEK(646)<>1 THEN POKE646,1
3275 L=LEN(R$):FOR JL=1 TO 20:FOR CR=1 TO 40-L:X=CR-1
3280 Y=JL+2:C$=MID$(B$(JL),CR,L)
3282 IF C$<>R$ THEN 3290
3285 YT=24:GOSUB2800:IC=X+1:YC=Y:CA=IC-1:CC=JL:C$=B$(CC):GOSUB3561:E=1:GOTO3296
3290 NEXT CR
3292 NEXT JL
3294 YT=24:GOSUB2800
3296 X=25:Y=24:GOSUB11000:PRINT"■CTRL+Y : AYUDA";
3298 X=1:Y=24:GOSUB11000:PRINT"PAGINA ";NS/20+1;:POKE 646,1
3300 IF E=1 THEN E=0:GOTO 3030
3305 GOTO 3010
3400 REM ------- B1$<<B$ -------
3410 FOR I=NS+1 TO NS+20:B1$(I)=B$(I-NS):NEXT I
3420 RETURN
3500 REM --- ENTRADA CAMPO ---
3552 C$=B$(CC):X=XC:Y=YC:IC=1
3555 GOSUB11000:PRINTC$:UL=XC+L0-1:CA=0
3560 X=XC+IC-1:IF X<XC THEN X=XC
3561 IF X>UL THEN X=UL
3562 IF R=1 THEN R=0:A0=17:IC=0:XC=0:X=0:B$(CC)=C$:RETURN
3563 IF CA=L0 THEN A0=17:B$(CC)=C$:RETURN
3564 GOSUB11000:KK=IC:IF IC=0 THEN KK=1
3565 GET A$:IF A$<>"" THEN 3580
3570 GOSUB11000:PRINT"■";:GOSUB11000:PRINTMID$(C$,KK,1);
3575 FOR T=1 TO 50:NEXT T:GOTO 3565
3580 A0=ASC(A$)
3582 IF A0=34 OR A0=252 OR A0=253 OR A0=254 THEN 3565
3585 IF A0=145 OR A0=17 THEN B$(CC)=C$:RETURN
3590 FOR K9=3 TO 8:IF A0=TF(K9) THEN B$(CC)=C$:RETURN
3592 NEXT K9
3595 IF A0=134 AND CC<>0 THEN GOSUB 3700:GOTO 3560
3597 IF A0=138 AND CC<>0 THEN GOSUB 3800:GOTO 3560
3600 IF A0=25 AND CC<>0 THEN B$(CC)=C$:GOSUB 13000:GOTO 3552
3605 IF A0=13 AND CC<>0 THEN A0=60:R=1:GOTO 3610
3606 IF A0<>13 THEN 3610
3607 C$=MID$(C$,1,IC):X=XC+IC-1
3608 A$=MID$(BU$,1,L0-IC+1)
3609 GOSUB11000:PRINTA$;:RETURN
3610 IF A0<>157 THEN 3620
3612 IF IC=0 AND CC=0 THEN 3617
3614 IF IC=0 THEN A0=145:X=40:IC=39:RETURN
3616 IC=IC-1:CA=CA-1
3617 X=X-1:IF X<XC THEN X=XC
3618 GOTO 3560
3620 IF A0<>29 THEN 3624
3621 IF A0=29 THEN IC=IC+1:CA=CA+1:X=X+1
3622 IF IC>L0 THEN IC=L0
3623 GOTO 3560
3624 CH$=CHR$(A0)
3625 PRINT"■";CH$;:C$=MID$(C$,1,IC-1)+CH$+MID$(C$,IC+1,39):IC=IC+1:X=X+1:CA=CA+
1
3626 IF CA=>L0 THEN CA=L0
3627 GOTO 3560
3640 REM --------------------------------
3700 REM * INSERCION *
3705 IF CA=L0 THEN A0=0:RETURN
3715 GET A$:IF A$="" THEN 3715
3717 A0=ASC(A$)
3720 IFA0=140ORA0=130RA0=134ORA0=145ORA0=17THENC$=C$+MID$(BA$,1,40-L):RETURN
3724 IF A0=136ORA0=25ORA0=157ORA0=29THENC$=C$+MID$(BA$,1,40-L):RETURN
3730 CH$=CHR$(A0):C$=MID$(C$,1,CA)+CH$+MID$(C$,CA+1,L0-CA-1):L=LEN(C$)
3735 A$=MID$(BA$,1,L0-L):A$=C$+A$:X=XC:Y=YC:GOSUB11000:PRINTA$:X=X+1:IC=IC+1:CA=
CA+1
3740 IF CA<L0 THEN 3715
3745 A0=17:RETURN
3800 REM * BORRADO *
3805 IF CA=L0 THEN A0=0:RETURN
3810 C$=MID$(C$,1,IC-1)+MID$(C$,IC+1,L0-IC-1)
3815 L=LEN(C$):A$=MID$(BA$,1,L0-L):A$=C$+A$:X=XC:Y=YC:GOSUB11000:PRINTA$
3820 C$=A$:RETURN
6000 REM * APERTURA FICHERO Y COMANDOS *
6002 OPEN 1,8,15
6005 IF OP=1 THEN OPEN2,8,3,"0:"+NF$+",L,"+CHR$(41):GOTO 6012
6010 OPEN 2,8,3,NF$
6012 INPUT#1,AD,XD$,CD,DD
6014 RETURN
6800 REM -- APUNTA AL REGISTRO #PA --
6805 R1=PA:R2=0
6810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
6815 PRINT#1,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
6820 RETURN
6900 REM -- CIERRA LOS CANALES --
6905 CLOSE1:CLOSE2
6910 RETURN
10000 REM ** DATOS **
10001 REM :
10005 REM ------ CADENAS AYUDA ------
10007 DATA" _1 = PAGINA ADELANTE "
10009 DATA" _2 = PAGINA ATRAS "
10010 DATA" _3 = INSERCION CARACTER "
10011 DATA" _4 = BORRADO CARACTER "," _5 = INSERCION LINEA "
10012 DATA" _6 = BORRADO LINEA "
10013 DATA" _7 = BUSCA PALABRA "
10015 DATA" _8 = RETORNO AL MENU + INTRODUCCION "
10016 DATA" CRSR UP,DOWN,RIGHT,LEFT = DESPLAZAN EL CURSOR "
10020 DATA134,138,140,133,137,135,139,136
10022 REM :
10025 REM ------ VOCI DEL MENU ------
10027 DATA"IESTION DISCO","¯SCRITURA","_IN DE TAREA"
10030 REM :
10035 REM ----- MENU GEST. DISCO ----
10040 DATA"¯REACION FICHERO","¯SCRITURA FICHERO"
10045 DATA"LECTURA FICHERO","_ETORNO AL MENU"
10999 REM
11000 REM * IMPRIMIR EN *
11001 REM
11005 POKE 211,X:POKE 214,Y
11010 SYS58732
11015 RETURN
11100 REM * PRINT AT *
11101 REM
11105 POKE 211,X1:POKE 214,Y1
11110 SYS58732
11115 RETURN
13000 REM -- AYUDA --
13002 HE$="":FOR JJ=1 TO 9:HE$=HE$+H$(JJ):NEXT JJ
13005 YT=24:GOSUB2800
13010 A$=BU$:POKE 198,0
13040 FOR I=1 TO 250:FORT=1TO100:NEXTT:CH$=MID$(HE$,I,1):IFCH$=""THENCH$=" "
```

```
13050 A$=RIGHT$(A$,37)+CH$:X=1:Y=24:GOSUB11000:PRINTA$;
13052 GET V$:IF V$<>"" THEN 13110
13100 NEXT I:GOTO 13040
13110 GOSUB 2900
13120 X=1:Y=24:GOSUB11000:PRINT"PAGINA ";NS/20+1;
13130 X=25:Y=24:GOSUB11000:PRINT"CTRL+Y : AYUDA";:POKE198,0:RETURN
40000 REM CONTROLES UNIDAD DISCO
40002 IF AD=0 OR AD=50 THEN RETURN
40005 IF AD=26 THEN B$="DISCO PROTEGIDO,IMPOSIBLE ESCRIBIR"
40010 IF AD=72 THEN B$="ESPACIO EN DISCO AGOTADO"
40012 IF AD=74 THEN B$="UNIDAD DISCO NO EN LINEA,COMPROBAR"
40014 IF AD=70 THEN B$="CANAL NO DISPONIBLE"
40015 IF AD=62 AND OP<>3 THEN RETURN
40020 IF AD=62 AND OP=3 THEN B$="FICHERO DATOS "+NF$+" NO ESTA"
40025 LU=LEN(B$):SP%=(40-LU)/2
40027 K$=""
40030 FOR P=1 TO SP%:K$=K$+CHR$(32):NEXT P
40035 X=0:Y=24:GOSUB11000:PRINTK$+B$+K$;:POKE 198,0:WAIT 198,1
40040 RUN
45000 REM CONTROL CARACTERES NO ACEPTABLES POR LA UNIDAD
45005 LS=LEN(B1$(I))
45010 FOR W=1 TO LS
45012 CF$=MID$(B1$(I),W,1)
45020 IF CF$=CHR$(44) THEN CF$=CHR$(253)
45025 IF CF$=CHR$(58) THEN CF$=CHR$(254)
45030 IF CF$=CHR$(59) THEN CF$=CHR$(252)
45035 IF CF$=CHR$(32) THEN CF$=CHR$(35)
45040 B1$(I)=LEFT$(B1$(I),W-1)+CF$+RIGHT$(B1$(I),LS-W)
45055 NEXT W
45060 RETURN
46000 REM SUSTITUCION CARACTERES NO ACEPTADOS POR LA UNIDAD
46005 LS=LEN(B1$(I))
46010 FOR W=1 TO LS
46012 CF$=MID$(B1$(I),W,1)
46020 IF CF$=CHR$(253) THEN CF$=CHR$(44)
46025 IF CF$=CHR$(254) THEN CF$=CHR$(58)
46030 IF CF$=CHR$(252) THEN CF$=CHR$(59)
46035 IF CF$=CHR$(35) THEN CF$=CHR$(32)
46040 B1$(I)=LEFT$(B1$(I),W-1)+CF$+RIGHT$(B1$(I),LS-W)
46055 NEXT W
46060 RETURN
```

Impresión

```
10 REM ***************************
12 REM *                         *
15 REM * IMPRESION TEXTOS C-64   *
17 REM *                         *
20 REM ***************************
25 M3=402:HO$="":PRINTCHR$(14)
27 SP$="":FORK=1TO80:SP$=SP$+CHR$(32):NEXTK
30 DIMB$(100),D$(20),XD(20),YD(20),XC(20),YC(20),LC(20),TP(20),B1$(M3),P(80)
35 NN=80:RR=40:NF=1
40 GOSUB500:X=1:Y=12:GOSUB11000:INPUT"NOMBRE DEL FICHERO";NF$
42 IFNF$=""ORNF$=CHR$(13)THEN40
43 IF LEN(NF$)>6 THEN 40
45 IF LEN(NF$)<6 THEN FORK=1TO6-LEN(NF$):NF$=NF$+CHR$(32):NEXT K
50 GOSUB 700
55 GOSUB 1300:IF OPZ=5 THENPRINTHO$:PRINTCHR$(142):POKE646,14:END
60 ON OPZ GOSUB 4100,4500,4700,9000
70 GOTO 55
500 PRINTHO$:X=0:Y=0:GOSUB11000:PRINT" .I.♥.          DICIONES ORUM, ♥.♠."
505 X=12:Y=Y+2:GOSUB11000:PRINT"IMPRESION TEXTOS"
510 RETURN
700 REM LECTURA DATOS
705 OPEN15,8,15
710 OPEN2,8,3,NF$
715 INPUT#15,AD,X$,CD,DD
720 IF AD<>0 THEN GOSUB 40000:RUN
725 PA=1:GOSUB 800:INPUT#2,A$
727 MX=VAL(LEFT$(A$,3)):PG=VAL(MID$(A$,4,2))
730 FOR I=1 TO MX
735 PA=I+1:GOSUB 800:INPUT#2,A$
737 GOSUB 25000
740 B1$(I)=A$
745 X=1:Y=23:GOSUB11000:PRINT"LECTURA LINEA :";I;
750 NEXT I:CLOSE15:CLOSE2
755 RETURN
800 REM PUNTERO
805 R1=PA:R2=0
810 IF R1>256 THEN R2=INT(PA/256):R1=R1-256*R2
815 PRINT#15,"P"CHR$(3+96)CHR$(R1)CHR$(R2)CHR$(1)
820 RETURN
1300 REM ** MENU **
1305 PRINTHO$:PRINTCHR$(14)
1310 RESTORE:FORI=1TO5:READVO$(I):NEXT
1315 GOSUB 500:X=10:Y=6
1317 FOR K=1 TO 5:Y=Y+2:GOSUB11000:PRINT"";K;"  ";VO$(K):NEXT K
1318 POKE 198,0
1320 X=11:Y=19:GOSUB11000:INPUT"VAL ";OPZ
1325 IF OPZ<=0 OR OPZ>5 THEN 1320
1330 RETURN
2500 REM
2530 L=LEN(Z$):IFL=0THENRETURN
2535 FORI=1TOL:IFMID$(Z$,I,1)<>" "THENI1=I:GOTO2560
2540 NEXTI:Z$="":RETURN
2560 FORI=LTO1STEP-1:IFMID$(Z$,I,1)<>" "THENI2=I:GOTO2590
2580 NEXT I
2590 K0=I2-I1+1:Z$=MID$(Z$,I1,K0):RETURN
4100 REM    JUSTIFICACION
4102 GOSUB 4500
4105 GOSUB 500
4107 KK=-(NN=80)*2-(NN=40)
4110 FOR IR=1 TO MX STEPKK
4115 X=1:Y=22:GOSUB11000:PRINT"JUSTIFICACION LINEA";IR
4120 Z$=B1$(IR):IF NN=80 THEN Z$=Z$+B1$(IR+1)
4130 GOSUB 2500:IF Z$<>""THEN GOSUB 4200
4135 B1$(IR)=LEFT$(Z$,40)
4137 IF C1$="<" AND I<=40 AND NN=80 THEN IR=IR-1:GOTO 4140
4138 IF NN=80 THEN B1$(IR+1)=MID$(Z$,41,40)
4140 NEXT IR:RETURN
4200 REM JUSTIFICA UNA LINEA
4210 L=LEN(Z$):BL=0:FORI=1TOL
4220 C1$=MID$(Z$,I,1):C2$=MID$(Z$,I+1,1)
4225 IF C1$="<" THEN RETURN
4230 IF C1$=" "AND C2$<>" "THEN BL=BL+1:P(BL)=I
4240 NEXT I:IFBL=0ORL=NNTHEN RETURN
4250 K=0:FORI=1TOBL
4260 Z$=MID$(Z$,1,P(I)+K)+" "+MID$(Z$,P(I)+K+1,NN)
4270 IFLEN(Z$)>=NN THENRETURN
4280 K=K+1:NEXT I
4290 K=0:FORI=1TOBL:K=K+1:P(I)=P(I)+K:NEXTI
4300 GOTO 4250
4500 REM PARAMETROS
4510 GOSUB 500
4520 X=1:Y=22:GOSUB11000:INPUT"NUMERO DE COLUMNAS (40/80) ";NN
4525 IF NN<>40 AND NN<>80 THEN 4520
4540 GOSUB500:X=1:Y=22:GOSUB11000:INPUT"LINEAS POR PAGINA";RR:IFRR<=0THEN4540
4580 GOSUB 500:X=1:Y=22:GOSUB11000:PRINT"NUMERACION (S/N)"
4582 GET A$:IF A$<>"N"ANDA$<>"/"ANDA$<>"S"ANDA$<>"♥"THEN4582
4590 IFA$="S"ORA$="♥"THENNF=1:RETURN
4592 NF=0:RETURN
4700 REM IMPRESION
4702 OPEN4,4
4705 GOSUB 4710:CMD4:PRINTCHR$(17):GOTO 4730
4710 GOSUB500:X=10:Y=12:GOSUB11000:PRINT"POSICIONAR EL PAPEL":X=9:Y=18:GOSUB110
00
4715 PRINT"  PULSAR UNA TECLA"
4720 POKE 198,0:WAIT 198,1
4725 PRINTHO$:X=13:Y=14:GOSUB11000:PRINT"E S P E R A R":RETURN
4730 II=0:PP=1
4735 REM
4736 KK=-(NN=80)*2-(NN=40)
4740 FOR I=1TO MXSTEPKK
4745 IF II>=RR THEN GOSUB 20000:II=0:PP=PP+1
4747 IF II=0ANDNF=1THENPRINT"PAGINA";PP:PRINT
4750 A$=B1$(I):IF NN=80 THEN A$=A$+B1$(I+1)
4751 IFA$=""ORA$=LEFT$(SP$,NN) THENPRINT:GOTO4800
4760 FORJ=1TONN:CH$=MID$(A$,J,1)
4770 IFCH$="<"THENR3=1:PRINT:II=II+1:R=1:GOTO4790
4775 IF CH$=" " AND R=1 THEN 4790
4780 R=0:PRINTCH$;
4790 NEXT J:IF R3=0 THEN PRINT:II=II+1
4800 R3=0:NEXT I:GOSUB 20000:IF OPZ=3 THEN PRINT#4:CLOSE4
4805 RETURN
9000 REM IMPRESION CON FUSION
```

```
9002 PRINTHO$:X=(40-LEN(VO$(OPZ)))/2:Y=2:GOSUB11000:PRINTVO$(OPZ)
9004 X=0:Y=15:GOSUB11000:PRINT"▮INTRODUCIR EL DISCO DE LAS DEFINICIONES"
9006 X=10:Y=17:GOSUB11000:PRINT"DE LA |♠♥¯ ¯¯ ¯♠|Γ♥"
9007 X=11:Y=23:GOSUB11000:PRINT"▮PULSAR UNA TECLA":POKE198,0:WAIT198,1:POKE198,0
9010 OPEN15,8,15,"I/O"
9012 OPEN2,8,3,"0:DEF0,SEQ,R"
9013 INPUT#15,AD,X$,CD,DD:IF AD<>0 THEN GOSUB40000:RETURN
9015 FOR I=1 TO 20
9020 INPUT#2,R$
9025 XD(I)=VAL(MID$(R$,1,2))
9030 YD(I)=VAL(MID$(R$,3,2))
9035 D$(I)=MID$(R$,5,10)
9040 XC(I)=VAL(MID$(R$,15,2))
9045 YC(I)=VAL(MID$(R$,17,2))
9050 LC(I)=VAL(MID$(R$,19,2))
9055 TP(I)=VAL(MID$(R$,21,2))
9060 NEXT I:CLOSE2:CLOSE15
9063 GR=1:FOR HH=1 TO 20:GR=GR+LC(HH):NEXT HH
9065 REM ELECCION DE LOS CAMPOS A IMPRIMIR
9300 PRINTHO$:FORI=1TO20:IFD$(I)=LEFT$(SP$,10)THEN9307
9305 X=XD(I):Y=YD(I):GOSUB11000:PRINTD$(I)
9307 NEXT I
9310 FORI=1TO20:IFTP(I)=0THEN9322
9320 X=XC(I):Y=YC(I):GOSUB11000:PRINT"▮"STR$(I)
9322 NEXT I
9325 X=0:Y=22:GOSUB11000:PRINT"▮INDICAR LOS CAMPOS A IMPRIMIR:"
9330 FORI=1TO20:IFTP(I)=0THEN9370
9340 X=0:Y=23:GOSUB11000:PRINT"¬AMPO N.▮";I;"▮? (▮▮▮/▮▮▮)";
9350 GETA$:IFA$="/"ORA$="N"THENTP(I)=0:GOTO9370
9360 IFA$<>"S"ANDA$<>"♥"THEN9350
9370 NEXT I
9380 GOSUB500:X=1:Y=11:GOSUB11000:INPUT"/OMBRE FICHERO (DE |♠♥¯ ¯¯ ¯♠|Γ♥)";NF$
9385 IF NF$=""ORNF$=CHR$(13)THEN9380
9390 OPEN15,8,15,"I/O"
9392 OPEN2,8,3,"0:"+NF$+",SEQ,R":INPUT#15,AD,X$,CD,DD:IFAD<>0THEN40000:RETURN
9400 INPUT#2,A$:MK=VAL(A$)
9410 FORI=1TOMK:GOSUB9600:B$(I)=AA$:NEXT I:CLOSE2:CLOSE15
9420 IR=1:LL=2:OPEN4,4:GOSUB 4710
9430 REM IMPRESION CAMPOS
9435 CMD4:PRINTCHR$(17);
9440 FORI=1TO20
9450 A$=MID$(B$(IR),LL,LC(I))
9460 IFTP(I)<>0 THENPRINTA$:I5=I5+1
9470 NEXTI:LL=2:PRINT
9480 GOSUB 4730:IR=IR+1
9510 IFIR>MKTHENPRINT#4:CLOSE4:RETURN
9520 I5=0:GOTO 9440
9600 REM FORMACION REGISTRO #I
9605 AA$="":FOR K9=1 TO GR
9610 GET#2,A$:AA$=AA$+A$
9615 NEXT K9:RETURN
10000 DATA"\USTIFICACION","¬ARAMETROS IMPRESION","\MPRESION"
10005 DATA"\MPRESION CON FUSION","_IN DE TAREA"
11000 REM
11005 POKE211,X:POKE214,Y
11010 SYS58732
11015 RETURN
20000 REM SALTO PAGINA
20003 OJ=66-II-NF*2-I5
20005 FOROI=1TOOJ:PRINT:NEXTOI:RETURN
20010 REM
25000 REM SUSTITUCION CARACTERES
25005 LS=LEN(A$)
25010 FOR K1=1 TO LS
25015 CT$=MID$(A$,K1,1):CT=ASC(CT$)
25020 IF CT=253 THEN CT$=CHR$(44)
25021 IF CT=254 THEN CT$=CHR$(58)
25022 IF CT=252 THEN CT$=CHR$(59)
25023 IF CT=35 THEN CT$=CHR$(32)
25030 A$=LEFT$(A$,K1-1)+CT$+RIGHT$(A$,LS-K1)
25035 NEXT K1:RETURN
40000 REM CONTROLES DISCO
40002 X1$="¯_Γ_ "
40005 IF AD=62 THEN X$=X1$+"EL FICHERO NO ESTA"
40006 IF AD=74 OR AD=70 THEN X$=X1$+"UNIDAD DISCO NO EN LINEA"
40007 IF AD=64 THEN X$=X1$+"TIPO DE FICHERO NO CORRESPONDIENTE"
40010 X1=INT((40-LEN(X$))/2):X=X1:Y=23
40015 GOSUB11000:PRINTX$
40020 GET R$:IF R$="" THEN 40020
40025 RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - COMMODORE 64

A0	= código de la tecla pulsada en escritura
CC	= contador de caracteres
C$	= cadena introducida (subrutina 3500)
X,Y	= coordenadas actuales
B$(*)	= matriz para los buffers de máscara vídeo
H$(*)	= cadenas ayuda
B1$(*)	= un elemento para cada línea de texto
MX	líneas totales
NP	= número de páginas
PG	= líneas por página
OP	= opción elegida en el menú principal
AA$	= cadena I/O con el disco
NS	= flag para página actual
L0	= longitud máxima para C$
LC(*)	= longitud del campo de introducción
XC(*), **YC(*)**	= coordenadas en el vídeo del campo de introducción
XD(*), **YD(*)**	= coordenadas en el vídeo de la instrucción
D$(*)	= descripciones en las máscaras vídeo
VO$(*)	= voces del menú principal
BL$	= blank
NF$	= nombre del fichero datos
NN	= número de columnas en la impresión
RR	= líneas por página en impresión



Tratamiento de textos/Versión Apple II

Tratamiento de textos

```
10   REM  ----------------------
20   REM   TRATAMIENTO DE TEXTOS
23   REM          FUENTE
25   REM        VERSION DOS
27   REM  ----------------------
30   ONERR  GOTO 40000
40   DIM B$(20):
     DIM B1$(400):
     MX = 20:
     NP = 1:
     PG = 20
50   K = 1:
     GOSUB 1300:
     IF OP = 4 THEN  TEXT :
     HOME :
     END
70   IF OP < > 3 THEN 100
80   PRINT D1$:
     PRINT D1$;"RUN IMPRESION FUENTE,D
     1"
100  ON OP GOSUB 2000,3000:
     GOTO 50
500  REM  ---------------------
530  TEXT :
     HOME :
     BP$ =  CHR$ (7):
     D1$ =  CHR$ (4):
     DR$ = "1":
     B$ = "          ":
     BA$ = B$ + B$ + B$ + B$
565  HE$(1) = "CTRL E.RETORNO AL MENU
     - CTRL I.INSERCION CARACTER - CTR
     L D.BORRADO CARACTER - CTRL.P.PAG
     INA ADELANTE - "
566  HE$(2) = "CTRL O.PAGINA ATRAS - C
     TRL L.INSERCION LINEA - CTRL K.BO
     RRADO LINEA - CTRL R.BUSQUEDA PAL
     ABRA - CTRL W,A,S,Z.MUEVEN EL CUR
     SOR"
570  HE$(2) = HE$(2) + "      PULSAR UN
     A TECLA  ":
     B$ = "":
     HTAB 1:
     VTAB 1:
     INVERSE :
     PRINT "E.G.S.              EDICION
     ES FORUM, S.A.":
     NORMAL :
     FC = 1:
     RETURN
1000 REM
1010 X1 = X - 1:
     Y1 = Y - 1:
     CH =  SCRN( X1,2 * Y1) + 16
      *  SCRN( X1,2 * Y1 + 1):
     HTAB X:
     VTAB Y:
     FLASH :
     IF FC = 0 THEN  INVERSE
1040 PRINT  CHR$ (CH);:
     HTAB X:
     NORMAL :
     RETURN
1100 REM
1110 X = X1 + 1:
     Y = Y1 + 1:
     NORMAL :
     HTAB X:
     VTAB Y:
     PRINT  CHR$ (CH);:
     HTAB X:
     RETURN
1200 REM
1210 A0 =  PEEK ( - 16384):
     IF A0 > 128 THEN  POKE  - 16368,0
     :
     A0 = A0 - 128:
     RETURN
1220 GOTO 1210
1300 REM  ---------------------
1330 GOSUB 500:
     RESTORE :
     FOR J = 1 TO K:
        READ NV:
        IF NV = 0 THEN  RETURN
1350    L = 0:
        FOR I = 1 TO NV:
           READ VO$(I):
           LL =  LEN (VO$(I)):
           IF LL > L THEN L = LL
1370    NEXT I:
     NEXT J:
     XI = (40 - L) / 2:
     YI = ((24 - NV * 2) / 2) + 1:
     X = XI - 2:
     Y = YI:
     FOR I = 1 TO NV:
        HTAB X:
        VTAB Y:
        PRINT I;" - ";VO$(I):
        Y = Y + 2:
     NEXT I:
     Y = Y + 1
1410 HTAB X:
     VTAB Y:
     PRINT "CUAL ? ";:
     GOSUB 1200:
     OP$ =  CHR$ (A0):
     OP =  VAL (OP$):
     IF OP < 1 OR OP > NV THEN
      PRINT BP$;:
     GOTO 1410
1440 FLASH :
     PRINT OP:
     NORMAL :
     FOR I = 1 TO 100:
     NEXT I:
     RETURN
2000 REM  ---------------------
2010 K = 2:
     GOSUB 1300:
     IF OP = 4 THEN  RETURN
2020 GOSUB 500:
     HTAB 5:
     VTAB 13:
     PRINT "NOMBRE FICHERO ? "
2030 XC = 23:
     YC = 13:
     LO = 6:
     CC = 0:
     GOSUB 3500:
     NF$ = C$:
     IF NF$ = "" THEN 2030
2045 LR = 40:
     GOSUB 6000:
     IF OP < > 1 THEN 2100
2047 GOSUB 500
2050 HTAB 5:
     VTAB 13:
     PRINT "NUMERO DE PAGINAS ? "
2060 XC = 27:
     LO = 2:
     GOSUB 3500:
     NP =  VAL (C$):
     IF NP < = 0 THEN 2060
2070 GOSUB 500:
     HTAB 5:
     VTAB 13:
     PRINT "LNEAS POR PAGINA ? "
2080 XC = 29:
     GOSUB 3500:
     PG =  VAL (C$):
     IF PG < = 0 THEN 2080
2090 MX = PG * NP:
     IF MX > 400 THEN  GOSUB 500:
     HTAB 1:
     VTAB 23:
     PRINT "MAX.400 LINEAS DE TEXTO.":
     GOTO 2050
2100 IF OP = 3 THEN 2170
2105 GOSUB 500:
     A$ = "CREO ":
     IF OP = 2 THEN A$ = "GRABO "
2110 FOR I = 1 TO MX:
        IF OP = 1 THEN B1$(I) = BA$
2125    HTAB 1:
        VTAB 23:
        PRINT A$;"LA LINEA ";I:
        AA$ = B1$(I):
        NR = I + 1:
        GOSUB 6400:
     NEXT I:
     IF OP = 1 THEN NR = 1:
     AA$ =  STR$ (MX) +  MID$ (BA$,1,3
      -  LEN ( STR$ (MX))) +  STR$ (PG
     ):
     GOSUB 6400
2160 GOSUB 6600:
     GOTO 2010
2170 GOSUB 500:
     NR = 1:
     GOSUB 6200:
     MX =  VAL ( LEFT$ (AA$,3)):
     PG =  VAL ( MID$ (AA$,4,2)):
     FOR I = 1 TO MX:
        NR = I + 1:
        GOSUB 6200:
        HTAB 1:
        VTAB 23:
        PRINT "LEO LA LINEA ";I:
        B1$(I) = AA$:
     NEXT :
     GOSUB 6600:
     GOTO 2010
2800 REM  ---------------------
2830 A$ =  LEFT$ (BA$,26):
     HTAB 1:
     VTAB YT:
     PRINT A$;:
     RETURN
3000 REM  ---------------------
3010 IC = 0:
     X = 1:
     XC = 1:
     CC = 1:
     LO = 40:
     XC = 1:
     FOR I = 1 TO 20:
        B$(I) = B1$(I + NS):
        HTAB XC:
        VTAB I + 2:
        IF B$(I) = "" THEN B$(I)
         = BA$
```

```
3012    PRINT B$(I);:
        NEXT I:
        HTAB 27:
        VTAB 24:
        PRINT "AYUDA. CTRL-Y";:
        HTAB 1:
        VTAB 24:
        PRINT "PAGINA ";NS / 20 + 1;:
        IF K < > 0 THEN CC = K:
        K = 0
3020    YC = CC + 2:
        GOSUB 3500
3030    IF AO = 5 THEN  GOSUB 3400:
        RETURN
3040    IF AO < > 26 THEN 3070
3050    CC = CC + 1:
        IF CC > 20 THEN 3105
3060    GOTO 3020
3070    IF AO < > 23 THEN 3100
3080    CC = CC - 1:
        IF CC < 1 THEN 3135
3090    GOTO 3020
3100    IF AO < > 16 THEN 3130
3105    GOSUB 3400:
        NS = NS + 20:
        IF NS > = MX THEN NS = NS
         - 20
3120    GOTO 3010
3130    RETURN
3135    GOSUB 3400:
        NS = NS - 20:
        IF NS < 0 THEN NS = 0
3150    GOTO 3010
3160    IF AO < > 12 THEN 3200
3165    GOSUB 3400:
        FOR I = MX - 1 TO NS + CC STEP
         - 1:
          B1$(I + 1) = B1$(I):
        NEXT I:
        B1$(NS + CC) = BA$:
        K = CC:
        GOTO 3010
3200    IF AO < > 11 THEN 3250
3205    GOSUB 3400:
        FOR I = NS + CC + 1 TO MX:
          B1$(I - 1) = B1$(I):
        NEXT I:
        B1$(MX) = BA$:
        K = CC:
        GOTO 3010
3250    IF AO < > 18 THEN 3020
3255    GOSUB 3400:
        HTAB 1:
        VTAB 24:
        PRINT "PALABRA ? ";:
        XC = 8:
        YC = 24:
        LO = 15:
        CC = 0:
        GOSUB 3500:
        R$ = C$:
        L =  LEN (R$):
        YT = 24:
        LO = 40:
        XC = 1:
        FOR J = 1 TO 20:
          FOR X = 1 TO 40 - L:
            Y = J + 2:
            GOSUB 1000:
            GOSUB 1100:
            C$ =  MID$ (B$(J),X,L):
            IF C$ = R$ THEN  GOSUB 2800
            :
            IC = X - 1:
            YC = Y:
            CC = J:
            C$ = B$(CC):
            GOSUB 3550:
            GOTO 3030
3310    NEXT X,J:
        GOSUB 2800:
        GOTO 3010
3400    REM  ---------------------
3410    FOR I = NS + 1 TO NS + 20:
          B1$(I) = B$(I - NS):
        NEXT I:
        RETURN
3500    REM  ---------------------
3510    IF CC = 0 OR IC = 40 THEN IC
         = 0:
        X = 1
3530    C$ = B$(CC):
        IF LO = 0 THEN  RETURN
3540    Y = YC:
        IF CC = 0 THEN X = XC
3545    IF IC = LO THEN AO = 26:
        B$(CC) = C$:
        RETURN
3547    IF R = 1 THEN R = 0:
        AO = 26:
        IC = 0:
        X = 1:
        XC = 1:
        B$(CC) = C$:
        RETURN
3550    GOSUB 1000:
        GOSUB 1200:
        GOSUB 1100
3560    IF AO = 5 OR AO = 23 OR AO
         = 26 OR AO = 16 OR AO = 15
         OR AO = 12 OR AO = 11 OR AO
         = 18 OR AO = 27 THEN B$(CC)
         = C$:
        RETURN
3565    IF AO = 9 AND CC < > 0 THEN
         GOSUB 3700:
        GOTO 3560
3567    IF AO = 4 AND CC < > 0 THEN
         GOSUB 3800:
        GOTO 3545
3570    IF AO = 25 AND CC < > 0 THEN
         GOSUB 13000:
        GOTO 3545
3580    IF AO = 13 AND CC < > 0 THEN AO
         = 60:
        R = 1:
        GOTO 3620
3585    IF AO < > 13 THEN 3620
3590    C$ =  MID$ (C$,1,IC):
        A$ =  MID$ (BA$,1,LO - IC):
        HTAB XC + IC:
        PRINT A$;:
        RETURN
3620    IF AO < > 1 THEN 3655
3630    IF IC = 0 AND CC = 0 THEN
         PRINT BP$;:
        GOTO 3545
3635    IF IC = 0 THEN AO = 23:
        X = 40:
        IC = 39:
        RETURN
3640    IC = IC - 1:
        X = X - 1:
        GOTO 3545
3655    IF AO = 0 THEN 3545
3660    IF AO = 19 THEN IC = IC + 1:
        X = X + 1:
        GOTO 3545
3670    CH$ =  CHR$ (AO):
        VTAB YC:
        PRINT CH$;:
        C$ =  MID$ (C$,1,IC) + CH$
         +  MID$ (C$,IC + 2,39):
        IC = IC + 1:
        X = X + 1:
        GOTO 3545
3700    REM  ---------------------
3722    IF IC = LO THEN AO = 0:
        RETURN
3730    FC = 0:
        GOSUB 1000:
        GOSUB 1200:
        GOSUB 1100:
        IF AO = 5 OR AO = 13 OR AO
         = 4 OR AO = 23 OR AO = 26
         OR AO = 18 OR AO = 27 OR AO
         = 25 OR AO = 1 OR AO = 19
         THEN C$ = C$ +  MID$ (BA$,1,40
         - L):
        FC = 1:
        RETURN
3750    CH$ =  CHR$ (AO):
        C$ =  MID$ (C$,1,IC) + CH$
         +  MID$ (C$,IC + 1,LO - IC
         - 1):
        L =  LEN (C$):
        A$ =  MID$ (BA$,1,LO - L):
        A$ = C$ + A$:
        HTAB XC:
        VTAB YC:
        PRINT A$:
        X = X + 1:
        IC = IC + 1:
        IF IC < LO THEN 3730
3790    FC = 1:
        AO = 26:
        RETURN
3800    REM  ---------------------
3830    IF IC = LO THEN  RETURN
3840    C$ =  MID$ (C$,1,IC) +  MID$ (C$,
        IC + 2,LO - IC - 1):
        L =  LEN (C$):
        A$ =  MID$ (BA$,1,LO - L):
        A$ = C$ + A$:
        HTAB XC:
        PRINT A$:
        C$ = C$ +  MID$ (BA$,1,40 - L):
        RETURN
6000    REM  ---------------------
6030    PRINT D1$:
        PRINT D1$;"OPEN";NF$ + ",D"
         + DR$;",L";LR + 2:
        PRINT D1$:
        RETURN
6200    REM  ---------------------
6230    PRINT D1$:
        PRINT D1$;"READ";NF$;",R";NR:
        INPUT AA$:
        PRINT D1$:
        RETURN
6400    REM  ---------------------
6430    PRINT D1$:
        PRINT D1$;"WRITE";NF$;",R";NR:
        PRINT  CHR$ (34) + AA$:
        PRINT D1$:
        RETURN
6600    REM  ---------------------
6630    PRINT D1$:
        PRINT D1$;"CLOSE";NF$:
        PRINT D1$:
        RETURN
10010   DATA 4:
        DATA "GESTION DISCO","ESCRITURA",
        "IMPRESION","FIN DE TAREA":
        DATA 4:
        DATA "GENERACION","ESCRITURA FICH
        ERO","LECTURA FICHERO","RETORNO A
        L MENU"
13000   REM  ---------------------
13030   YT = 24:
        GOSUB 2800
13035   FOR H = 1 TO 2:
          HE$ = HE$(H)
13040     FOR I = 1 TO  LEN (HE$):
            FOR T = 1 TO 200:
            NEXT T:
            CH$ =  MID$ (HE$,I,1):
```

```
       IF CH$ = "" THEN CH$
          = " "
13060     A$ =  RIGHT$ (A$,37)
          + CH$:
       HTAB 1:
       VTAB 24:
       PRINT A$;:
       A0 =  PEEK ( - 16384):
       IF A0 > 128 THEN 13110
13100 NEXT 1,H:
   GOTO 13035
13110 GOSUB 2800:
   HTAB 27:
   VTAB 24:
   PRINT "AYUDA. CTRL-Y";:
   HTAB 1:
   VTAB 24:
   PRINT "PAGINA ";NS / 20 + 1;:
   POKE  - 16368,0:
   RETURN
40000 REM --------------------
40010 GOSUB 500:
   INVERSE :
   HTAB 1:
   VTAB 13:
   E =  PEEK (222):
   IF E = 5 THEN  PRINT "EL FICHERO
   NO EXISTE ";
40030 IF E = 8 THEN  PRINT "COMPROBAR L
   A UNIDAD DISCO. ";
40040 IF E = 9 THEN  PRINT "DISCO LLENO
   . ";
40050 IF E = 6 THEN  PRINT "INTRODUCIR
   DISCO PROGRAMAS. ";
40060 IF E = 4 THEN  PRINT "DISCO PROTE
   GIDO. ";
40070 NORMAL :
   HTAB 1:
   VTAB 15:
   PRINT "PULSAR UNA TECLA";
   GOSUB 1200:
   GOTO 50
```

Impresión

```
10   REM   ----------------------
20   REM   TRATAMIENTO DE TEXTOS
22   REM          IMPRESION
25   REM         VERSION DOS
30   REM   ----------------------
35   ONERR  GOTO 40000
40   DIM B1$(402),B$(130),P(80),XD(20)
     ,YD(20),XC(20),YC(20),LC(20),TP(2
     0),D$(20):
     NN = 80:
     RR = 40:
     NF = 1
50   GOSUB 500:
     HTAB 8:
     VTAB 13:
     PRINT "NOMBRE FICHERO ? "
60   XC = 24:
     YC = 13:
     LO = 6:
     CC = 0:
     GOSUB 3500:
     NF$ = C$:
     IF NF$ = "" THEN 60
80   LR = 40:
     GOSUB 6000:
     NR = 1:
     GOSUB 6200:
     MX =  VAL ( LEFT$ (AA$,3)):
     PG =  VAL ( MID$ (AA$,4,2)):
     GOSUB 500:
     FOR I = 1 TO MX:
        NR = I + 1:
        GOSUB 6200:
        HTAB 1:
        VTAB 23:
        PRINT "LEO LA LINEA ";I:
        B1$(I) = AA$:
     NEXT I:
     GOSUB 6600
120  K = 1:
     GOSUB 1300:
     IF OP = 5 THEN  TEXT :
     HOME :
     END
140  ON OP GOSUB 4100,4500,4700,9000:
     IR = 0:
     GOTO 120
500  REM  ---------------------
530  TEXT :
     HOME :
     BP$ =  CHR$ (7):
     D1$ =  CHR$ (4):
     DR$ = "1":
     B$ = "          ":
     BA$ = B$ + B$ + B$ + B$:
     HTAB 1:
     VTAB 1:
     INVERSE :
     PRINT "E.G.S              EDICIONE
     S FORUM. S.A.":
     NORMAL :
     FC = 1:
     RETURN
1000 REM  ---------------------
1010 X1 = X - 1:
     Y1 = Y - 1:
     CH =  SCRN( X1,2 * Y1) + 16
      *  SCRN( X1,2 * Y1 + 1):
     HTAB X:
     VTAB Y:
     FLASH :
     IF FC = 0 THEN  INVERSE
1040 PRINT  CHR$ (CH);:
     HTAB X:
     NORMAL :
     RETURN
1100 REM  ---------------------
1110 X = X1 + 1:
     Y = Y1 + 1:
     NORMAL :
     HTAB X:
     VTAB Y:
     PRINT  CHR$ (CH);:
     HTAB X:
     RETURN
1200 REM  ---------------------
1210 A0 =  PEEK ( - 16384):
     IF A0 > 128 THEN  POKE  - 16368,0
     :
     A0 = A0 - 128:
     RETURN
1220 GOTO 1210
1300 REM  ---------------------
1330 GOSUB 500:
     RESTORE :
     FOR J = 1 TO K:
        READ NV:
        IF NV = 0 THEN  RETURN
1350    L = 0:
        FOR I = 1 TO NV:
           READ VO$(I):
           LL =  LEN (VO$(I)):
           IF LL > L THEN L = LL
1370    NEXT I:
     NEXT J:
     XI = (40 - L) / 2:
     YI = ((24 - NV * 2) / 2) + 1:
     X = XI - 2:
     Y = YI:
     FOR I = 1 TO NV:
  HTAB X:
  VTAB Y:
  PRINT I;" - ";VO$(I):
  Y = Y + 2:  •
     NEXT I:
     Y = Y + 1
1410 HTAB X:
     VTAB Y:
     PRINT "CUAL ?";:
     GOSUB 1200:
     OP$ =  CHR$ (A0):
     OP =  VAL (OP$):
     IF OP < 1 OR OP > NV THEN
      PRINT BP$;:
     GOTO 1410
1440 FLASH :
     PRINT OP:
     NORMAL :
     FOR I = 1 TO 500:
     NEXT I:
     RETURN
2500 REM  ---------------------
2530 L =  LEN (Z$):
     IF L = 0 THEN  RETURN
2535 FOR I = 1 TO L:
        C =  ASC ( MID$ (Z$,I,1)):
        IF C <  > 32 THEN I1 = I:
        GOTO 2560
2550 NEXT I:
     Z$ = "":
     RETURN
2560 FOR I = L TO 1 STEP  - 1:
        C =  ASC ( MID$ (Z$,I,1)):
        IF C <  > 32 THEN I2 = I:
        GOTO 2590
2580 NEXT I
2590 KO = I2 - I1 + 1:
     Z$ =  MID$ (Z$,I1,KO):
     RETURN
2800 REM  ---------------------
2830 A$ =  LEFT$ (BA$,39):
     HTAB 1:
     VTAB YT:
     PRINT A$;:
     RETURN
3500 REM  ---------------------
3510 IF CC = 0 OR IC = 40 THEN IC
      = 0:
     X = 1
3530 C$ = B$(CC):
     IF LO = 0 THEN  RETURN
3540 Y = YC:
     IF CC = 0 THEN X = XC
3545 IF IC = LO THEN A0 = 26:
     B$(CC) = C$:
     RETURN
3550 GOSUB 1000:
     GOSUB 1200:
     GOSUB 1100
3560 IF A0 = 5 OR A0 = 23 OR A0
      = 26 OR A0 = 16 OR A0 = 15
      OR A0 = 12 OR A0 = 11 OR A0
      = 18 OR A0 = 27 THEN B$(CC)
      = C$:
     RETURN
3580 IF A0 = 13 AND CC <  > 0 THEN A0
      = 60:
     GOTO 3620
3585 IF A0 <  > 13 THEN 3620
3590 C$ =  MID$ (C$,1,IC):
     A$ =  MID$ (BA$,1,LO - IC):
     HTAB XC + IC:
     PRINT A$;:
     RETURN
3620 IF A0 <  > 1 THEN 3655
3630 IF IC = 0 AND CC = 0 THEN
      PRINT BP$;:
     GOTO 3545
3635 IF IC = 0 THEN A0 = 23:
```

```
       X = 40:
       IC = 39:
       RETURN
3640   IC = IC - 1:
       X = X - 1:
       GOTO 3545
3655   IF AO = 0 THEN 3545
3660   IF AO = 19 THEN IC = IC + 1:
       X = X + 1:
       GOTO 3545
3670   CH$ =  CHR$ (AO):
       VTAB YC:
       PRINT CH$;:
       C$ =  MID$ (C$,1,IC) + CH$
        +  MID$ (C$,IC + 2,39):
       IC = IC + 1:
       X = X + 1:
       GOTO 3545
4100   REM  ---------------------
4102   GOSUB 4500:
       KK = (NN = 80) * 2 + (NN = 40)
4105   GOSUB 500:
       FOR IR = 1 TO MX STEP KK:
          HTAB 1:
          VTAB 23:
          PRINT "JUSTIFICO LA LINEA ";IR
          :
          Z$ = B1$(IR):
          IF NN = 80 THEN Z$ = Z$
           + B1$(IR + 1)
4110      GOSUB 2500:
          IF Z$ <  > "" THEN  GOSUB 4200

4135      B1$(IR) =  LEFT$ (Z$,40)
4137      IF C1$ = "<" AND I <  = 40
           AND NN = 80 THEN IR = IR
           - 1:
          GOTO 4140
4138      IF NN = 80 THEN B1$(IR + 1)
           =  MID$ (Z$,41,40)
4140   NEXT IR:
       PRINT D1$:
       PRINT D1$;"PR#0":
       RETURN
4200   REM  ---------------------
4210   L =  LEN (Z$):
       BL = 0:
       FOR I = 1 TO L:
          C1$ =  MID$ (Z$,I,1):
          C2$ =  MID$ (Z$,I + 1,1):
          IF C1$ = "<" THEN  RETURN
4230      IF C1$ = " " AND C2$ <
           > " " THEN BL = BL + 1:
          P(BL) = I
4240   NEXT I:
       IF BL = 0 OR L = NN THEN  RETURN

4250   K = 0:
       FOR I = 1 TO BL:
          Z$ =  MID$ (Z$,1,P(I) + K)
           + " " +  MID$ (Z$,P(I)
           + K + 1,NN):
          IF  LEN (Z$) >  = NN THEN
           RETURN
4280      K = K + 1:
       NEXT I:
       K = 0:
       FOR I = 1 TO BL:
          K = K + 1:
          P(I) = P(I) + K:
       NEXT I:
       GOTO 4250
4500   REM  --------------------+
4510   GOSUB 500:
       HTAB 1:
       VTAB 23:
       PRINT "NUMERO DE COLUMNAS ? (40/8
       0)"
4530   XC = 30:
       YC = 23:
       LO = 2:
       GOSUB 3500:
       NN =  VAL (C$):
       IF NN <  > 40 AND NN <  > 80
        THEN 4530
4550   GOSUB 500:
       HTAB 1:
       VTAB 23:
       PRINT "LINEAS POR PAGINA ? "
4560   XC = 21:
       GOSUB 3500:
       RR =  VAL (C$):
       IF RR <  = 0 THEN 4560
4580   GOSUB 500:
       B$(0) = "":
       HTAB 1:
       VTAB 23:
       PRINT "NUMERACION ? (S/N)"
4590   XC = 20:
       LO = 1:
       GOSUB 3500:
       IF C$ <  > "S" AND C$ <  > "N"
        THEN 4590
4600   NF = 0:
       IF C$ = "S" THEN NF = 1
4610   RETURN
4700   REM  ---------------------
4705   GOSUB 4710:
       GOTO 4735
4710   GOSUB 500:
       HTAB 1:
       VTAB 22:
       PRINT "PONER LA IMPRESORA EN LINE
       A ":
       GET R$
4712   PRINT D1$:
       PRINT D1$;"PR#1":
       YT = 23:
       GOSUB 2800:
       PRINT  CHR$ (9);"80N":
       PRINT  CHR$ (12);:
       RETURN
4735   II = 0:
       PP = 1:
       KK = (NN = 80) * 2 + (NN = 40):
       FOR I = 1 TO MX STEP KK:
          IF II > RR THEN  PRINT
           CHR$ (12);:
          II = 0:
          PP = PP + 1
4747      IF II = 0 AND NF = 1 THEN
           PRINT "PAGINA ";PP:
          PRINT
4750      A$ = B1$(I):
          IF NN = 80 THEN A$ = A$
           + B1$(I + 1)
4755      BC$ = BA$ + BA$:
          IF A$ =  LEFT$ (BC$,NN)
           OR A$ = "" THEN  PRINT :
          GOTO 4800
4760      FOR J = 1 TO NN:
             CH$ =  MID$ (A$,J,1)
4770         IF CH$ = "<" THEN R3
              = 1:
             PRINT :
             R = 1:
             GOTO 4790
4775         IF CH$ = " " AND R = 1
              THEN 4790
4780         R = 0:
             PRINT CH$;
4790      NEXT J:
          IF R3 = 0 THEN  PRINT
4800      II = II + 1:
          R3 = 0:
       NEXT I:
       IF IR = 0 THEN  PRINT D1$:
       PRINT D1$;"PR#0"
4810   RETURN
6000   REM  ---------------------
6030   PRINT D1$:
       PRINT D1$;"OPEN";NF$ + ",D"
        + DR$;",L";LR + 2:
       PRINT D1$:
       RETURN
6200   REM  ---------------------
6230   PRINT D1$:
       PRINT D1$;"READ";NF$;",R";NR:
       INPUT AA$:
       PRINT D1$:
       RETURN
6600   REM  ---------------------
6630   PRINT D1$:
       PRINT D1$;"CLOSE";NF$:
       PRINT D1$:
       RETURN
9000   REM  ---------------------
9110   GOSUB 500:
       HTAB 1:
       VTAB 11:
       PRINT "INTRODUCIR EL DISCO DATOS
       DE LA":
       PRINT "BASE DE DATOS":
       HTAB 10:
       VTAB 21:
       PRINT "DESPUES PULSAR UNA TECLA":
       GET A$:
       NF$ = "DEFC":
       LR = 26:
       GOSUB 6000:
       FOR I = 1 TO 20:
          NR = I:
          GOSUB 6200
9150      XD(I) =  VAL ( MID$ (AA$,1,2))
          :
          YD(I) =  VAL ( MID$ (AA$,3,2))
          :
          D$(I) =  MID$ (AA$,5,10):
          XC(I) =  VAL ( MID$ (AA$,15,2)
          ):
          YC(I) =  VAL ( MID$ (AA$,17,2)
          ):
          LC(I) =  VAL ( MID$ (AA$,19,2)
          ):
          TP(I) =  VAL ( MID$ (AA$,21,2)
          ):
       NEXT I:
       GOSUB 6600:
       GOSUB 500:
       FOR I = 1 TO 20:
          IF D$(I) =  LEFT$ (BA$,10)
           OR D$(I) = "" THEN 9290
9280      HTAB XD(I) + 1:
          VTAB YD(I) + 1:
          PRINT D$(I)
9290   NEXT I:
       FOR I = 1 TO 20:
          IF TP(I) = 0 THEN 9320
9310      HTAB XC(I) + 1:
          VTAB YC(I) + 1:
          PRINT " ";I
9320   NEXT I:
       HTAB 1:
       VTAB 23:
       PRINT "INDICA LOS CAMPOS A IMPRIM
       IR.":
       FOR I = 1 TO 20:
          IF TP(I) = 0 THEN 9370
9340      HTAB 1:
          VTAB 24:
          PRINT "CAMPO N. ";I;"? (S/N)";

9350      GET A$:
          IF A$ <  > "S" AND A$ <
           > "N" THEN 9350
9360      IF A$ = "N" THEN TP(I) = 0
9370   NEXT I:
```

```
      GOSUB 500:
      HTAB 1:
      VTAB 12:
      PRINT "NOMBRE FICHERO (DE LA BASE
       DATOS)";:
      INPUT NF$:
      LR = 128:
      GOSUB 6000:
      NR = 1:
      GOSUB 6200:
      MK =  VAL (AA$):
      FOR I = 1 TO MK:
         NR = I + 1:
         GOSUB 6200:
         B$(I) = AA$:
      NEXT I:
      GOSUB 6600:
      IR = 1:
      LL = 2:
      GOSUB 4710
9430  REM  ---------------------
9435  PRINT :
      FOR I = 1 TO 20:
         A$ =  MID$ (B$(IR),LL,LC(I)):
         LL = LL + LC(I):
         IF TP(I) <  > 0 THEN  PRINT A$

9490  NEXT I:
      LL = 2:
      PRINT :
      GOSUB 4735:
      IR = IR + 1:
      IF IR > MX THEN  PRINT D1$:
      PRINT D1$;"PR#0":
      PRINT D1$:
      RETURN
9520  GOTO 9430
10010 DATA 5:
      DATA "JUSTIFICACION","PARAMETROS"
      ,"IMPRESION","IMPRESION CON FUSIO
      N","FIN DE TAREA"
40000 GOSUB 500:
      INVERSE :
      HTAB 1:
      VTAB 13:
      E =  PEEK (222):
      IF E = 5 THEN  PRINT "EL FICHERO
      NO EXISTE.";
40030 IF E = 8 THEN  PRINT "COMPROBAR L
      A UNIDAD DE DISCO. ";
40040 IF E = 9 THEN  PRINT "DISCO LLENO
      . ";
40050 IF E = 6 THEN  PRINT "INTRODUCIR
      DISCO PROGRAMAS. ";
40060 IF E = 4 THEN  PRINT "DISCO PROTE
      GIDO. ";
40070 NORMAL :
      HTAB 1:
      VTAB 15:
      PRINT "PULSAR UNA TECLA":
      GOSUB 1200:
      GOTO 50
```

RELACION DE LAS PRINCIPALES VARIABLES - APPLE II

B$(∗) = matriz para los buffers de máscara vídeo
H$(∗) = cadenas ayuda
B1$(∗) = un elemento para cada línea de texto
MX = líneas totales
NP = número de páginas
PG = líneas por página
OP = opción elegida en el menú principal
AA$ = cadena I/O con el disco
NS = flag para página actual
A0 = código de la tecla pulsada en escritura
CC = contador de caracteres
C$ = cadena introducida (subrutina 3500)
X,Y = coordenadas actuales
L0 = longitud máxima para C$
LC(∗) = longitud del campo de introducción
XC(∗), **YC(∗)** = coordenadas en el vídeo del campo de introducción
XD(∗), **YD(∗)** = coordenadas en el vídeo de la descripción
D$(∗) = descripciones en las máscaras video
VO$(∗) = voces del menú principal
BL$ = blank
NF$ = nombre del fichero datos
NN = número de columnas en la impresión
RR = líneas por página en impresión

Tratamiento de textos/Versión Philips VG 8010

Tratamiento de textos

```
10    REM ---------------------
12    REM TRATAMIENTO DE TEXTOS
13    REM     Version MSX
14    REM ---------------------
15    LOCATE,,0
20    CLEAR17000:DEFINTA-Z:SCREEN0:WIDTH40
25    KEYOFF:KEY1,CHR$(25):KEY2,CHR$(5):KEY3,CHR$(4):KEY4,CHR$(9):KEY5,CHR$(12):KEY6,CHR$(11):KE
Y7,CHR$(15):KEY8,CHR$(16):KEY9,CHR$(18)
35    M3=400
40    DIM B$(20),B1$(M3):MX=20:NP=1:PG=20
50    K=1:GOSUB 1300
60    IF OP=4 THEN CLS:END
70    IF OP<>3 THEN 100
80    GOSUB 500:LOCATE4,10:PRINT"Posicionar la cinta PROGRAMAS":LOCATE13,13:PRINT"y apretar PLAY
":LOCATE10,20:PRINT"despues pulsar una tecla"
82    IF INKEY$="" THEN 82
85    GOSUB 500:LOCATE13,10:PRINT"E S P E R A R":LOCATE1,16:PRINT"al terminar la carga teclear R
UN"
90    CLOAD "IMPRESION"
91    STOP
100   ONOPGOSUB 2000,3000
110   GOTO50
500   REM ----------------
530   CLS:BP$="O5L32ABC"
540   HE$="F2:vuelve al menu - F3:borra un caracter - F4:insercion caracteres - F5:insercion lin
eas - F6:borrado linea - F7:pagina atras - F8: pagina adelante - F9:busqueda palabra - PULSAR UN
 TECLA -"
580   LOCATE0,0:PRINT"TRAT.TEX. E.G.S.    Ediciones Forum S.A.":LOCATE0,1:FOR R=1 TO 40:PRINTCHR
$(1);CHR$(&H57);:NEXT R
590   RETURN
1000  REM --- ENCIENDE CURSORE ---
1010  LOCATEX-1,Y-1,1
1020  RETURN
1100  REM --- APAGA CURSOR ---
1110  LOCATEX-1,Y-1,0
1120  RETURN
1200  REM ESPERA CARACTER ---
1210  A$=INKEY$:IF A$="" OR A$="," THEN 1210
1220  AO=ASC(A$):RETURN
1300  REM --- MENU ---
1330  GOSUB 500:RESTORE:FOR J=1 TO K:READ NV:IF NV=0 THEN RETURN
1350  L=0:FORI=1TONV:READVO$(I):LL=LEN(VO$(I)):IFLL>LTHENL=LL
1370  NEXTI,J:XI=(40-L)/2:YI=((24-NV*2)/2)+1:X=XI-2:Y=YI:FORI=1TONV:LOCATEX-1,Y-1:PRINT I;"- ";V
O$(I):Y=Y+2:NEXTI:Y=Y+1
1410  LOCATEX-1,Y-1:PRINT"CUAL ?";:GOSUB1200:OP$=CHR$(AO):OP=VAL(OP$):IF OP<1 ORO P>NV THEN PLAY
 BP$:GOTO 1410
1440 PRINT OP:FOR I=1 TO 1000:NEXT I:RETURN
2000  REM --- gestion cinta ---
2010  K=2:GOSUB 1300:IF OP=4 THEN RETURN
2020  IF OP<>1 THEN 2100
2030  GOSUB 500
2035  LOCATE10,12:PRINT"NUMERO PAGINAS ?"
2040  XC=27:YC=13:CC=0:LO=2:GOSUB3500:NP=VAL(C$):IF NP<=0THEN 2040
2050  GOSUB500:LOCATE8,12:PRINT"LINEAS POR PAGINA ?"
2060  XC=29:GOSUB 3500:PG=VAL(C$):IFPG<=0THEN2060
2070  MX=PG*NP:IF MX>M3 THEN GOSUB 500:LOCATE0,22:PRINT"MAX";M3;"LINEAS DE TEXTO.":GOTO 2035
2080  GOTO 2010
2100  REM ---------------
2120  GOSUB500:LOCATE7,12:PRINT"NOMBRE FICHERO ?"
2130  XC=23:YC=13:LO=6:CC=0:GOSUB 3500
2140  NF$=C$:IF NF$="" THEN 2130
2150  GOSUB 500:IF OP=2 THENT$="REC y PLAY" ELSE T$="    PLAY"
2160  LOCATE2,10:PRINT"Posicionar la cinta datos y apretar":LOCATE14,12:PRINTT$
2165  LOCATE10,20:PRINT"DESPUES PULSAR UNA TECLA"
2170  IF INKEY$="" THEN 2170
2180  GOSUB 500:LOCATE13,11:PRINT"E S P E R A R"
2190  IF OP=3 THEN 2300
2200  OPEN"CAS:■"FOROUTPUTAS#1:A$="":FOR I=1 TO 6:PRINT#1,A$:NEXT:CLOSE#1:REM espacio cinta
2210  OPEN"CAS:"+NF$ FOROUTPUTAS#1
2220  A$="*"+SPACE$(5):MID$(A$,2,3)=RIGHT$(STR$(MX),3):MID$(A$,5,2)=RIGHT$(STR$(PG),2):PRINT#1,A
$
2230  FOR I=1 TO MX:LOCATE0,22:PRINT"GRABO LA LINEA";I
2240  A$="*"+B1$(I):PRINT#1,A$
2250  NEXT I:CLOSE#1:GOTO 2010
2300  REM --- LECTURA ---
2310  OPEN"CAS:"+NF$ FORINPUTAS#1
2320  INPUT#1,A$:MX=VAL(MID$(A$,2,3)):PG=VAL(MID$(A$,5,2))
2330  FOR I=1 TO MX:LOCATE0,22:PRINT"LEO LA LINEA";I
2340  INPUT#1,A$:A$=MID$(A$,2,40)
2350  B1$(I)=A$+SPACE$(40-LEN(A$))
2360  NEXTI:CLOSE#1:GOTO2010
2800  REM ----------------
2830  A$=SPACE$(26):LOCATE0,YT-1:PRINTA$;:RETURN
2900  REM --- ELIMINA ESPACIOS ---
2930  Z2$="":A=LEN(Z$):FORI=1TOA:Z1$=MID$(Z$,I,1):IFZ1$<>" "THENZ2$=Z2$+Z1$
2950  NEXTI:Z$=Z2$:RETURN
3000  REM --- escritura ---
3010  IC=0:X=1:XC=1:CC=1:LO=40:FOR I=1 TO 20:B$(I)=B1$(I+NS):LOCATEXC-1,I+1:IFB$(I)="" THEN B$(I
)=SPACE$(40)
3012  PRINTB$(I);:NEXTI:LOCATE26,23:PRINT"AYUDA: F1";:LOCATE0,23:PRINT"PAGINA";NS/20+1;
3015  IF K<>0 THEN CC=K:K=0
3020  YC=CC+2:GOSUB 3500
3030  IF AO=5 THEN GOSUB 3400:RETURN
3040  IF AO<>31 THEN 3070
3050  CC=CC+1:IF CC>20 THEN 3105
3060  GOTO 3020
3070  IF AO<>30 THEN 3100
3080  CC=CC-1:IFCC<1THEN3135
3090  GOTO3020
3100  IFAO<>16THEN3130
3105  GOSUB3400
3110  NS=NS+20:IFNS>=MXTHENNS=NS-20
3120  GOTO3010
3130  IFAO<>15THEN3160
3135  GOSUB 3400
3140  NS=NS-20:IFNS<0THENNS=0
3150  GOTO3010
3160  IFAO<>12THEN3200
3165  GOSUB 3400
3170  FOR I=MX-1 TO NS+CC STEP-1
3180  B1$(I+1)=B1$(I):NEXTI
3190  B1$(NS+CC)=SPACE$(40):K=CC:GOTO 3010
3200  IF AO<>11 THEN 3250
3205  GOSUB 3400
3210  FOR I=NS+CC+1 TO MX
3220  B1$(I-1)=B1$(I)
3230  NEXT I:B1$(MX)=SPACE$(40)
3240  K=CC:GOTO 3010
3250  IF AO<>18 THEN 3020
3255  GOSUB 3400
3260  LOCATE0,23:PRINT"PALABRA ? ";:XC=8:YC=24:LO=15:CC=0:GOSUB3500:R$=C$
3270  L=LEN(R$):YT=24:LO=40:XC=1
3280  FOR J=1 TO 20:FOR X=1 TO 40-L
3290  Y=J+2:GOSUB 1000:GOSUB 1100
3300  C$=MID$(B$(J),X,L):IF C$=R$ THEN GOSUB 2800:IC=X-1:YC=Y:CC=J:C$=B$(CC):LOCATE0,23:PRINT"PA
GINA";NS/20+1;:GOSUB 3550:GOTO 3030
3310  NEXT X,J:GOSUB 2800:GOTO 3010
3400  REM --- B1$<<B$ ---
3410  FOR I=NS+1 TO NS+20:B1$(I)=B$(I-NS):NEXT
3420  RETURN
3500  REM --------------------
3510  IF CC=0 OR IC=40 THEN IC=0:X=1
3530  C$=B$(CC):IF LO=0 THEN RETURN
3540  Y=YC:IF CC=0 THEN X=XC
3545  IF IC=LO THEN AO=31:B$(CC)=C$:RETURN
3547  IF R=1 THEN R=0:AO=31:IC=0:X=1:XC=1:B$(CC)=C$:RETURN
3550  GOSUB 1000:GOSUB 1200:GOSUB 1100
3560  IF AO=5 OR AO=30 OR AO=31 OR AO=16 OR AO=15 OR AO=12 OR AO=11 OR AO=18 OR AO=27 THEN B$(CC
)=C$:RETURN
3565  IF AO=9 AND CC<>0 THEN GOSUB 3700:GOTO 3560
3567  IF AO=4 AND CC<>0 THEN GOSUB 3800:GOTO 3545
3570  IF AO=25 AND CC<>0 THEN GOSUB 13000:GOTO 3545
```

```
3580 IF A0=13 AND CC<>0 THEN A0=60:R=1:GOTO 3620
3585 IF A0<>13 THEN 3620
3590 C$=MID$(C$,1,IC):A$=SPACE$(LO-IC):LOCATEXC+IC-1,YC-1:PRINTA$;:RETURN
3620 IF A0<>29 THEN 3655
3630 IF IC=0 AND CC=0 THEN PLAYPB$:GOTO 3545
3635 IF IC=0 THEN A0=23:X=40:IC=39:RETURN
3640 IC=IC-1:X=X-1:GOTO 3545
3655 IF A0=0 THEN 3545
3660 IF A0=28 THEN IC=IC+1:X=X+1:GOTO 3545
3670 CH$=CHR$(A0):LOCATE,YC-1:PRINTCH$;:C$=MID$(C$,1,IC)+CH$+MID$(C$,IC+2,39):IC=IC+1:X=X+1:GOT
O 3545
3700 REM --------------------
3722 IF IC=LO THEN A0=0:RETURN
3730 GOSUB 1000:GOSUB 1200:GOSUB 1100:IF A0=5 OR A0=13 OR A0=4 OR A0=30 OR A0=31 OR A0=18 OR A0
=27 OR A0=25 OR A0=29 OR A0=28 OR A0=9 THEN C$=C$+SPACE$(40-L):RETURN
3750 CH$=CHR$(A0):C$=MID$(C$,1,IC)+CH$+MID$(C$,IC+1,LO-IC-1):L=LEN(C$):A$=SPACE$(LO-L):A$=C$+A$
:LOCATEXC-1,YC-1:PRINTA$:X=X+1:IC=IC+1:IFIC<LOTHEN3730
3790 A0=31:RETURN
3800 REM --------------------
3830 IF IC=LO THEN RETURN
3840 C$=MID$(C$,1,IC)+MID$(C$,IC+2,LO-IC-1):L=LEN(C$):A$=SPACE$(LO-L):A$=C$+A$:LOCATEXC-1:PRINT
A$:C$=C$+SPACE$(40-L):RETURN
10000 REM --- DATOS ---
10010 DATA 4
10020 DATA "gestion cinta","escritura","impresion","fin de tarea"
10030 DATA 4
10040 DATA "generacion","escritura fichero","lectura fichero","vuelve al menu"
13000 REM --- AYUDA ---
13030 YT=24:GOSUB2800
13040 FORI=1TO205:FORT=1TO30:NEXTT:CH$=MID$(HE$,I,1)
13060 A$=RIGHT$(A$,37)+CH$:LOCATE0,23:PRINTA$;:IFINKEY$<>""THEN13110
13100 NEXTI:GOTO13040
13110 GOSUB2800
13130 LOCATE0,23:PRINT"PAGINA";NS/20+1;TAB(26)"AYUDA: F1   ";:RETURN
```

Impresión

```
10   REM ---------------------
20   REM TRATAMIENTO DE TEXTOS
30   REM      IMPRESION
35   REM ---------------------
40   SCREEN0:KEYOFF:WIDTH40:CLEAR17000:DEFINTA-Z
50   M3=402:DIMB$(130),D$(20),XD(20),YD(20),XC(20),YC(20),LC(20),TP(20),B1$(M3),P(80):NN=80:RR=
40:NF=1:BP$="O5L32EFG"
55   GOSUB 500:X=9:Y=12:GOSUB11000:INPUT"NOMBRE FICHERO ";NF$
60   IF LEN(NF$)>6 THEN 55
70   IF NF$="" THEN 55
80   N$="DATOS":GOSUB700
85   OPEN"CAS:"+NF$ FORINPUTAS#1
90   INPUT#1,A$:MX=VAL(MID$(A$,2,3)):PG=VAL(MID$(A$,5,2))
100  GOSUB 500:FOR I=1 TO MX
110  INPUT#1,A$:B1$(I)=MID$(A$,2,40):LOCATE0,24:PRINT"LEO LA LINEA ";I;
120  NEXTI:CLOSE#1
130  GOSUB 1300
140  IF OP=5 THEN CLS:END
150  ON OP GOSUB 4100,4500,4700,9000
160  GOTO 130
500  CLS:X=0:Y=0:GOSUB11000:PRINT"TRAT.TEX.-impr. E.G.S.  Edic. Forum,S.A.":Y=1:GOSUB 11000:FOR
 R=1 TO 40:PRINTCHR$(1);CHR$(&H57);:NEXT R
510  RETURN
700  REM --- POSICIONA CINTA ---
710  GOSUB500:LOCATE9-LEN(N$)/2,10:PRINT"Posicionar la cinta ";N$:LOCATE12,14:PRINT"y apretar P
LAY"
720  LOCATE10,18:PRINT"Despues pulse una tecla"
730  IFINKEY$="" THEN 730
740  GOSUB 500:LOCATE13,11:PRINT"E S P E R A R"
750  RETURN
1300 REM --- MENU ---
1310 GOSUB 500:LOCATE15,5:PRINT"MENU":LOCATE15,6:FOR R=1 TO 4:PRINTCHR$(1);CHR$(&H57);:NEXT R:P
RINT
1320 LOCATE8,10:PRINT"1) JUSTIFICACION"
1330 LOCATE8,12:PRINT"2) PARAMETROS IMPRESION"
1340 LOCATE8,14:PRINT"3) IMPRESION"
1350 LOCATE8,16:PRINT"4) IMPRESION CON FUSION"
1360 LOCATE8,18:PRINT"5) fin de tarea"
1370 LOCATE11,21:PRINT"Cual ?"
1375 OP$=INKEY$:IF OP$<"1" OR OP$>"5" THEN 1375
1380 OP=VAL(OP$)
1390 RETURN
2500 REM -----------------------
2530 L=LEN(Z$):IF L=0 THEN RETURN
2535 FOR I=1 TO L:IF MID$(Z$,I,1)<>" "THENI1=I:GOTO 2560
2550 NEXT I:Z$="":RETURN
2560 FOR I=L TO 1STEP-1:IF MID$(Z$,I,1)<>" "THENI2=I:GOTO 2590
2580 NEXT I
2590 KO=I2-I1+1:Z$=MID$(Z$,I1,KO):RETURN
2800 REM --- BORRA LINEA YT ---
2830 A$=SPACE$(39):X=0:Y=YT:GOSUB 11000:PRINT A$;:RETURN
2900 REM --- ELIMINA ESPACIOS ---
2930 Z2$="":A=LEN(Z$):FORI=1TOA:Z1$=MID$(Z$,I,1):IFZ1$<>" "THENZ2$=Z2$+Z1$
2950 NEXT I:Z$=Z2$:RETURN
4100 REM --- JUSTIFICACIO ---
4102 GOSUB 4500
4105 GOSUB 500
4107 KK=-(NN=80)*2-(NN=40)
4110 FOR IR=1 TO MX STEPKK
4115 X=0:Y=22:GOSUB 11000:PRINT"JUSTIFICO LA LINEA ";IR
4120 Z$=B1$(IR):IF NN=80 THEN Z$=Z$+B1$(IR+1)
4130 GOSUB 2500:IF Z$<>"" THEN GOSUB 4200
4135 B1$(IR)=LEFT$(Z$,40)
4137 IF C1$="<" AND I<=40 AND NN=80 THEN IR=IR-1
4138 IF NN=80 THEN B1$(IR+1)=MID$(Z$,41,40)
4140 NEXT IR:RETURN
4200 REM --- JUSTIFICA UNA LINEA ---
4210 L=LEN(Z$):BL=0:FOR I=1 TO L
4220 C1$=MID$(Z$,I,1):C2$=MID$(Z$,I+1,1)
4225 IF C1$="<" THEN RETURN
4230 IF C1$=" " AND C2$<>" " THEN BL=BL+1:P(BL)=I
4240 NEXT I:IF BL=0 OR L=NN THEN RETURN
4250 K=0:FOR I=1 TO BL
4260 Z$=MID$(Z$,1,P(I)+K)+" "+MID$(Z$,P(I)+K+1,NN)
4270 IF LEN(Z$)>=NN THEN RETURN
4280 K=K+1:NEXT I
4290 K=0:FOR I=1 TO BL:K=K+1:P(I)=P(I)+K:NEXT I
4300 GOTO 4250
4500 REM --- PARAMETROS IMPRESION ---
4510 GOSUB 500
4520 X=0:Y=22:GOSUB11000:INPUT"NUMERO DE COLUMNAS ";NN
4525 IF NN<>40 AND NN<>80 THEN 4520
4540 GOSUB 500:X=0:Y=22:GOSUB 11000:INPUT"LINEAS POR PAGINA ";RR:IF RR<=0 THEN 4540
4580 GOSUB 500:X=0:Y=22:GOSUB 11000:PRINT"NUMERACION (S/N)"
4582 A$=INKEY$:IF A$<>"N" AND A$<>"n" AND A$<>"s" AND A$<>"S" THEN 4582
4590 IF A$="s" OR A$="S" THEN NF=1 ELSE NF=0
4600 RETURN
4700 REM --- IMPRESION ---
4705 GOSUB 4710:GOTO 4730
4710 GOSUB 500:X=9:Y=12:GOSUB 11000:PRINT"POSICIONAR EL PAPEL":X=10:Y=18:GOSUB 11000:PRINT"DESP
UES PULSE UNA TECLA"
4720 IFINKEY$="" THEN 4720
4725 GOSUB 740:RETURN
4730 II=0:PP=1
4736 KK=-(NN=80)*2-(NN=40)
4740 FOR I=1 TO MX STEP KK
4745 IF II>=RR THEN LPRINTCHR$(12):II=0:PP=PP+1
4747 IFII=0 AND NF=1 THEN LPRINT"PAGINA ";PP:LPRINT
4750 A$=B1$(I):IFNN=80THENA$=A$+B1$(I+1)
4751 IF A$="" OR A$=SPACE$(NN) THEN LPRINT:II=II+1:GOTO 4800
4760 FOR J=1 TO NN:CH$=MID$(A$,J,1)
4770 IF CH$="<" THEN LPRINT:II=II+1:R3=1:R=1:GOTO 4790
4775 IF CH$=" " AND R=1 THEN 4790
4780 R=0:LPRINTCH$;
4790 NEXT J:IF R3=0 THEN LPRINT:II=II+1
4800 R3=0:NEXT I:LPRINTCHR$(12):RETURN
9000 REM --------------------
9100 REM impresion con fusion
9110 REM   lectura FORMATO
9120 REM --------------------
9130 CLS:LOCATE 3,8:PRINT"Rebobinar la cinta DEFINICIONES":LOCATE11,11:PRINT"y apretar PLAY.":L
OCATE8,20:PRINT"Despues pulsar una tecla."
9140 IF INKEY$=""THEN 9140
9150 GOSUB 740:OPEN"CAS:DEFC"FORINPUTAS#1
9160 FOR I=1 TO 20
9170 INPUT#1,R$
9180 XD(I)=VAL(MID$(R$,1,2))
9190 YD(I)=VAL(MID$(R$,3,2))
```

```
9200  D$(I)=MID$(R$,5,10)
9210  XC(I)=VAL(MID$(R$,15,2))
9220  YC(I)=VAL(MID$(R$,17,2))
9230  LC(I)=VAL(MID$(R$,19,2))
9240  TP(I)=VAL(MID$(R$,21,2))
9250  NEXTI:CLOSE#1
9270  REM -----------------------
9280  REM ELECCION DE LOS CAMPOS
9290  REM -----------------------
9300  CLS:FOR I=1 TO 20:IF D$(I)=SPACE$(10) THEN 9307
9305  X=XD(I):Y=YD(I):GOSUB 11000:PRINTD$(I)
9307  NEXT I
9310  FOR I=1 TO 20:IF TP(I)=0 THEN 9322
9320  X=XC(I):Y=YC(I):GOSUB 11000:PRINTSTR$(I)
9322  NEXT I
9325  X=0:Y=22:GOSUB 11000:PRINT"INDICA LOS CAMPOS A IMPRIMIR:"
9330  FOR I=1 TO 20:IF TP(I)=0 THEN 9370
9340  X=0:Y=23:GOSUB 11000:PRINT"Campo N.";I;"? (S/N)";
9350  A$=INKEY$:IF A$="N" OR A$="n" THEN TP(I)=0:GOTO 9370
9360  IF A$<>"s" AND A$<>"S" THEN 9350
9370  NEXT I
9380  GOSUB 500:X=1:Y=11:GOSUB11000:PRINT"NOMBRE FICHERO (de la BASE de DATOS) ";:INPUT NF$
9390  N$="DATOS":GOSUB 700:OPEN"CAS:"+NF$FORINPUTAS#1:INPUT#1,A$:MK=VAL(A$)
9410  FOR I=1 TO MK:INPUT#1,B$(I):NEXT:CLOSE#1
9420  IR=1:LL=2:GOSUB 4710
9430  REM --- IMPRESION CAMPOS ---
9435  LPRINT
9440  FOR I=1 TO 20
9450  A$=MID$(B$(IR),LL,LC(I))
9460  LL=LL+LC(I)
9480  IF TP(I)<>0 THEN LPRINT A$
9490  NEXTI:LL=2:LPRINT
9500  GOSUB 4730:IR=IR+1
9510  IF IR>MK THEN RETURN
9520  GOTO 9430
11000 REM --- posiciona cursor ---
11010 LOCATE X,Y:RETURN
```

RELACION DE LAS PRINCIPALES VARIABLES - PHILIPS VG 8010

B$(*)	= matriz para los buffers de máscara vídeo
H$(*)	= cadenas ayuda
B1$(*)	= un elemento por cada línea de texto
MX	= líneas totales
NP	= número de páginas
PG	= líneas por página
OP	= opción elegida en el menú principal
AA$	= cadena I/O con la cinta
NS	= flag para página actual
A0	= código de la tecla pulsada en escritura
CC	= contador de caracteres
C$	= cadena introducida (subrutina 3500)
X,Y	= coordenadas actuales
L0	= longitud máxima para C$
LC(*)	= longitud del campo de introducción
XC(*), **YC(*)**	= coordenadas en el vídeo del campo de introducción
XD(*), **YD(*)**	= coordenadas en el vídeo de la descripción
D$(*)	= descripciones en las máscaras vídeo
VO$(*)	= voces del menú principal
BL$	= blank
NF$	= nombre del fichero datos
NN	= número de columnas en impresión
RR	= líneas por página en impresión

Extensiones y modificaciones

El modo de empleo y la propia finalidad de este tipo de programa están muy ligados a las necesidades específicas del usuario y, por tanto, las eventuales implantaciones dependen estrechamente del ámbito en el que debe trabajar el programa.

La orientación general que se ha dado al procedimiento es en el ámbito del trabajo de secretaría y, por tanto, dedicado a la preparación de correspondencia. Por este motivo se ha incluido la función de impresión con fusión, que permite la preparación de cartas y circulares de manera automática para distintas personas. Un segundo aspecto, muy importante, es la aplicación para la redacción de documentos técnicos o administrativos. En estos sectores, para tener una buena elasticidad de empleo, hay que implantar funciones específicas y, a veces, adaptadas a las particulares necesidades del usuario.

Los dos sectores de aplicación se analizan a continuación por separado, para así poder subrayar sus aspectos principales. Sin embargo, los temas que siguen no se exponen con detalle, sino que constituyen solamente algunas indicaciones para el desarrollo del análisis. A pesar de ello, su traducción en términos de instrucciones no presenta unas excesivas dificultades, porque puede obtenerse adaptando las subrutinas y los módulos empleados en los programas anteriores.

Aplicación en el sector técnico. Este tipo de empleo se caracteriza por:

1 / Presencia de dibujos o gráficos
2 / Tablas, eventualmente con desarrollo de cálculos
3 / Escasa utilización de la Base de Datos

Los puntos anteriores presentan, para este sector de aplicación, la necesidad de enlazar el programa con el Tablero electrónico y el programa de Gráficos.

Para realizar de manera automática estos tipos de conexión se presentan notables dificultades.

La solución más sencilla, que además no necesita modificaciones de programación, consiste en pasar de uno a otro programa de manera manual.

Para todos los procedimientos, esta solución ofrece todas las potencialidades, pero implica un empleo farragoso. La solución de mejor compromiso consiste en incluir algunas subrutinas del Tablero electrónico o de los Gráficos en el procedimiento del Tratamiento de textos.

Su activación puede vincularse a una tecla funcional y, por tanto, desde el punto de vista del procedimiento, pueden tratarse de manera análoga a las funciones ya presentes en el Tratamiento de textos.

No obstante, en la práctica, esta implantación no es tan sencilla como parece, puesto que la primera dificultad aparece con los nombres de las variables y con los números de líneas.

Por motivos de homogeneidad se han utilizado los mismos nombres de las variables en los procedimientos. Esta elección, útil para facilitar la comprensión de los distintos temas, es un obstáculo para la integración de partes de procedimientos diferentes. Sin embargo, no se trata de dificultades conceptuales, sino sólo formales, porque su solución tan sólo requiere una atenta lectura de los listados y un tedioso trabajo de variación de los nombres de las variables.

En cambio, las modificaciones de carácter sustancial son bastante más difíciles y siempre son necesarias en estos tipos de implantaciones. Para poner de manifiesto algunos aspectos, supóngase que se desea integrar en el Tratamiento de textos una sola columna del Tablero electrónico.

La primera modificación a realizar es la reducción al valor 1 del número máximo de columnas previstas en el Tablero electrónico.

Esta modificación tiene como consecuencia inmediata la necesidad de eliminar los desplazamientos horizontales (o sea las traslaciones horizontales de la pantalla). Además, como el Tratamiento de textos está organizado lógicamente por páginas, también hay que eliminar la traslación vertical, reduciendo el número de líneas al valor que puede contener la pantalla. Como complicación adicional, este valor deberá ser parametrizado para no vincular la zona de desarrollo de las tablas a una posición precisa en la pantalla. En la integración de funciones gráficas probablemente se derivarán problemáticas más complejas. Desde el punto de vista funcional, el problema puede resolverse borrando la pantalla (el texto no se pierde, puesto que queda en el buffer) y presentando el gráfico. En realidad, esta solución sólo es aparente, puesto que si bien permite presentar el texto y los gráficos en la pantalla, no realiza una verdadera integración, ni puede desarrollarla.

Aplicaciones en el sector administrativo. En este caso, los gráficos no suelen ser necesarios, a no ser marginalmente, mientras que se requiere un uso más intensivo de la Base de Datos, y también algunas necesidades de cálculo.

Sin embargo, hay que tener en cuenta que, en muchas aplicaciones, el desarrollo de los cálculos puede delegarse a la propia Base de Datos, por lo que en los casos más sencillos, la adaptación del procedimiento a este aspecto particular de aplicación se reduce a un enlace más completo sólo con la Base de Datos.

En principio, esta aplicación no presenta dificultades particulares porque puede derivarse de la impresión con fusión ya incluida en el procedimiento.

Epílogo

Hacia la inteligencia artificial

A nadie se le escapa la gran importancia que la informática tiene en las transformaciones del mundo actual. La informática junto a las telecomunicaciones se están constituyendo en el sistema nervioso central de nuestra sociedad. Resulta, pues, de gran utilidad considerar, aunque sea brevemente, la evolución histórica de la informática, sus tendencias actuales y su perspectiva de futuro, si queremos comprender todos estos cambios que tanto nos afectan.

El desarrollo de la informática ha sido rapidísimo, comparado con el de otras tecnologías. En sus cuarenta años de historia, su capacidad se ha multiplicado por diez cada cinco años. Pensemos que si los automóviles hubiesen mejorado en precio y calidad como lo han hecho los ordenadores, hoy tendríamos coches por la diezmilésima parte de su precio actual, con velocidades de hasta 1.000 km/h, gastando un litro de gasolina cada 10.000 km. y sin apenas averías durante 10.000 años.

Que este desarrollo de la informática haya sido tan rápido y espectacular no significa que haya sido fácil. Se han tenido que afrontar muchos desafíos y habrá que afrontar muchos más para mantener este ritmo de crecimiento. Por ejemplo, en el aspecto científico-técnico de la informática el esfuerzo es enorme, se investigan a la vez muchas posibilidades de progreso, de las cuales sólo algunas tendrán éxito. Como consecuencia, en el terreno social, resulta difícil encontrar buenos profesionales en las diferentes especialidades informáticas y hay que afrontar, pues, el problema de su formación o de una reconversión desde otras actividades. Por otra parte, los cambios son constantes dentro de la industria informática: cada tres años aparecen nuevos productos que superan la mitad de los existentes. Finalmente, en el campo comercial, la guerra actual entre Japón y Estados Unidos por la conquista de los mercados mundiales puede dejar fuera de juego a Europa por muy poco que ésta se descuide.

Para descubrir la evolución histórica de la informática, es útil considerar en ésta tres aspectos o componentes fundamentales. En primer lugar la evolución rápida de los materiales informáticos –el hardware, en inglés– cuya tecnología ha sido históricamente la primera en consolidarse a finales de los años setenta. En segundo lugar, el avance más dificultoso de la programación de ordenadores –el software–, cuya importancia se acentúa en la década de los setenta. Y en tercer lugar, las estructuras y sistemas de la información o de los datos, que están adquiriendo un papel propio y relevante en la actual década de los ochenta.

En la evolución de los materiales informáticos se suelen distinguir diferentes etapas, llamadas generaciones de ordenadores. La primera generación (1946-52) la constituyen los ordenadores de válvulas de vacío. El ENIAC (1946), por ejemplo, capaz de realizar 5.000 sumas por segundo, con una memoria de 20 números de 10 cifras, tenía 18.000 válvulas, pesaba 30 toneladas, ocupaba el piso entero de una escuela, tenía un consumo elevadísimo y había que cambiarle diariamente dos o tres válvulas fundidas. Hoy día, cualquier calculadora programable de bolsillo es mucho más potente que el ENIAC. La invención del transistor marca un avance fundamental y abre paso a la segunda generación de ordenadores (1953-65): las nuevas máquinas transistorizadas son mucho más pequeñas, más rápidas y más fiables. La tercera generación (1963-72) se caracteriza por la incorporación de circuitos integrados: en lugar de usar transistores aislados para después conectarlos, todos estos componentes electrónicos y sus conexiones son colocados conjuntamente sobre una pastilla de silicio mediante técnicas fotográficas. La cuarta generación (1973), la de los ordenadores actuales, se distingue de la anterior por la gran miniaturización o fuerte reducción en el tamaño de los circuitos integrados: un millón de elementos lógicos sobre una superficie de 2 milímetros cuadrados. En esta épo-

ca, también aparece el microprocesador, un único circuito integrado –un chip– que incorpora todos los elementos y funciones del ordenador. Resumiendo: en dos décadas la microelectrónica ha conseguido que los materiales informáticos fuesen rápidos, fiables y baratos, poniendo el microordenador al alcance de todo el mundo.

Sin embargo, por bueno que sea el material de que esté hecho el ordenador, éste de por sí sólo no sirve de mucho, porque para actuar necesita programas. Son los programas los que indican al ordenador lo que tiene que hacer exactamente. Los ordenadores son a la informática lo que los instrumentos musicales a la música y la programación o software son las partituras. Si el ordenador es o no una herramienta flexible y fácil de usar depende también de los programas que tenga en su memoria. Pero desgraciadamente, de los programas existentes actualmente menos de la mitad son buenos o tienen la calidad suficiente; la programación de ordenadores no ha sido ni es tarea fácil.

Para solucionar los problemas que se le plantean, el hombre ha desarrollado gran variedad de métodos propios, con un lenguaje y una lógica bien adaptados a cada problema. Con la aparición del ordenador el objetivo perseguido es que éste nos ayude en la solución de nuestros problemas; para ello hemos de dar al ordenador estos métodos y procedimientos de resolución. Pero desgraciadamente, la organización, el funcionamiento, el lenguaje y la lógica básicos de los ordenadores son relativamente elementales y alejados del proceder y lenguaje humanos. Para poder usar la máquina hay pues que eliminar esta distancia inicial hombre-máquina. Esto admite sólo dos soluciones. Primera, la más inmediata y práctica en espera de máquinas mejores, es que el hombre se adapte a la máquina. Y, desgraciadamente, esto ha sido así durante demasiado tiempo. La segunda solución posible, y la más deseable, ha sido ir acercando la máquina al hombre hasta hacer que el uso del ordenador sea una experiencia asequible a todo el mundo. En esta dirección, por una parte, ya en los años cincuenta, se logra que la máquina pueda entender lenguajes más humanos que el lenguaje binario de ceros y unos. Estos lenguajes permiten formular al ordenador problemas ya sea científicos (lenguaje Fortran) ya sea administrativos (lenguaje Cobal). Para ello hay que darle previamente al ordenador unos programas llamados compiladores que traducen los programas escritos por el hombre en estos lenguajes de alto nivel, al lenguaje binario ejecutable por la máquina. Estos progresos constituyen lo que se ha llamado la primera generación del software.

Además de afrontar la poca docilidad de las máquinas y la poca expresividad de los lenguajes de ordenador, la programación de ordenadores ha tenido que sufrir los efectos de la rapidísima evolución de los materiales informáticos o hardware. Cuando aún no se había aprovechado el 20% de las posibilidades de programación de una máquina ya aparecía otra más potente y más barata, a la cual había que adaptar los programas. Debido, pues, a la dificultad propia de la programación y a la necesidad de adaptar los programas a nuevas máquinas, a nuevas situaciones y aplicaciones, se vio clara la necesidad de establecer una metodología de la programación. El resultado fue lo que se ha denominado programación estructurada, que preconiza la construcción de programas como colecciones muy estructuradas de pequeños programas. Es la segunda generación del software de los años sesenta.

En la década de los setenta se desarrolla la tercera generación del software, constituida por nuevos sistemas operativos y nuevas herramientas para el desarrollo de programas. Y al mismo tiempo que progresa la tendencia a estructurar los programas, se siente la necesidad de estructurar los datos o informaciones con que se trabaja. Si en una etapa anterior los datos se consideraban embebidos en los programas, ahora el concepto de estructuras de datos o informaciones empieza a adquirir relieve por sí mismo. Así aparecen lo que se ha llamado bases de datos, que no son más que estructuras de la información. Por ejemplo, la estructuración de ficheros para tener fácil y rápido acceso a las informaciones que éstos contienen.

En la actualidad la microelectrónica mantiene un ritmo de avance parecido al de las épocas anteriores. Es de prever que a finales de siglo los componentes electrónicos de los ordenadores serán entre 100 y 1.000 veces mejores que los actuales, en cuanto a la relación coste/efectividad y tamaño/efectividad. Sin embargo, a partir de los años noventa serán necesarias nuevas técnicas, pues con las actuales se habrá alcanzado ya el límite de tamaño mínimo. Las memorias de masas –los discos magnéticos–, que ya han venido mejorando en un factor de 4 cada 5 años, continuarán mejorando a un ritmo parecido. Por otra parte la tecnología de los videodiscos permitirá almacenar y manipular cantidades enormes de información, integrando los textos y las imágenes en movimiento. La utilización de fibras ópticas en el interior de los ordenadores y entre ordenadores es otra línea de progreso. Aparecerán también nuevos componentes con nuevas capacidades: por ejemplo, matrices de sensores con capacidad de procesamiento y memoria integrada. El modelo o arquitectura de ordenador que ha dominado hasta hoy, llamado modelo de Von Neuman, está llegando al límite de su velocidad y potencia. Ya no es suficiente para las nuevas aplicaciones de la informática. Este modelo está limitado por su funcionamiento secuencial y ejecuta una instrucción detrás de otra; esto le impide ejecutar más de 10 instrucciones por segundo. Para avanzar más allá de este límite son necesarias máquinas no-secuenciales; máquinas capaces de ejecutar mu-

chas instrucciones en paralelo mediante diferentes procesadores interconectados. Para ello son necesarios también nuevos modelos o procedimientos de manipulación de informaciones en paralelo y los lenguajes de programación deberán poder expresar estos nuevos procedimientos. Y siguiendo esta evolución, el hardware del futuro –el transputador– ya no será un conjunto de operaciones en paralelo –como el cubo cósmico del MIT– sino que se concebirá todo el sistema como un flujo dinámico de datos o informaciones, gobernado por la misma disponibilidad de los datos.

Por otra parte, la informática y las telecomunicaciones convergen actualmente en lo que se ha llamado teleinformática o telemática: ordenadores interconectados a través de la red de comunicaciones pueden acceder a diferentes tipos de servicios informáticos (Teletex, Videotex, etc.). En la valoración de los microordenadores actuales ya se tiene en cuenta su facilidad para conectarse a las redes de comunicación. En un futuro podriámos tener integrados, en lo que se podría llamar teleordenador, los servicios del ordenador, del teléfono, de la televisión...

Disponer de un ordenador más dócil y amistoso que los actuales es ya una necesidad que se acentuará en los años venideros. Ya han aparecido en el mercado algunos ordenadores que mediante gráficos o iconos simulan el entorno de trabajo del hombre, facilitándole que escoja, en cada momento, la herramienta adecuada al problema planteado. El hombre dispone de todo lo que ve representado en la pantalla mediante gráficos. En un futuro sería deseable lograr que la máquina admitiese los medios de expresión típicamente humanos –lenguaje corriente hablado y escrito, imágenes gráficas, etc.– aunque fuere de forma restringida.

La misma programación de ordenadores ha progresado al admitir nuevos lenguajes, que en lugar de indicar al ordenador lo que éste tiene que hacer, le describen o especifican de forma completa el problema a resolver y dejan que sea el mismo ordenador el que determine cómo resolver el problema. Por ejemplo, la denominación «programación lógica» –el PROLOG es uno de sus lenguajes– permite formular explícitamente al ordenador los conocimientos necesarios para resolver un problema.

La mayoría de las líneas de progreso de la informática han sido una respuesta a las necesidades prácticas presentadas. Pero al mismo tiempo han existido desde siempre corrientes más teóricas, más especulativas. Entre ellas es de destacar la denominada inteligencia artificial, cuyo objetivo es utilizar los conceptos y las herramientas básicas de la informática para simular y comprender mejor la inteligencia humana, y así poder construir máquinas cada vez más inteligentes. Durante casi treinta años una pequeña comunidad de investigadores ha intentado con éxito diverso programar ordenadores para hacer de ellos sistemas o herramientas inteligentes de resolución de problemas. A mitad de los años setenta, después de dos décadas de progreso pesadamente lento, los investigadores de la inteligencia artificial llegaron a una conclusión fundamental sobre el comportamiento inteligente en general: requiere una tremenda cantidad de conocimientos, que la gente da a menudo por supuestos, pero que deben aportarse al ordenador con todo detalle. Por ejemplo, entender las expresiones del lenguaje corriente, incluso las más sencillas, requiere amplios conocimientos del contexto, sobre quien habla y el mundo en general, que están fuera de la capacidad de los ordenadores actuales. La única solución consiste en restringir o delimitar dicho contexto en cada caso. No es de extrañar, pues, que los programas con más éxito, los llamados sistemas expertos, son los que resuelven problemas en un dominio bien delimitado y de conocimientos fácilmente expresables en forma de reglas empíricas. Por ejemplo, las reglas empíricas utilizadas en todo tipo de diagnóstico, ya sea de enfermedades o de averías mecánicas; las reglamentaciones legislativas, las reglas de diseño y configuración de ordenadores, etc.

Los sistemas expertos han sido una de las primeras aplicaciones de la inteligencia artificial con éxito industrial y comercial. Otras áreas de aplicación, cuya experimentación se ha mantenido hasta ahora dentro de los laboratorios, pasarán a ser también aplicaciones industriales. Entre éstas, por su demanda social, son de destacar la traducción automática restringida y el resumen de documentos; la ayuda inteligente a la concepción, ingeniería y producción asistida por ordenador (CAD/CAM); el reconocimiento de textos hablados o escritos y su comprensión en dominios restringidos; la visión artificial y los robots inteligentes capaces de coordinar dinámicamente su percepción, sus acciones; la enseñanza asistida inteligentemente por ordenador; sistemas inteligentes de automatización de oficinas, etc.

Para afrontar estos desafíos de la inteligencia artificial es necesario un nuevo tipo de ordenador no-secuencial y programable mediante lenguajes lógicos de especificación de problemas. A este nuevo tipo de ordenador los japoneses lo han llamado ordenador de la quinta generación. Y lo han concebido para que sea fácilmente utilizable tanto en el trabajo como en casa: por tanto su comportamiento ha de ser fácilmente comprensible por el hombre; habrá de disponer de las máximas facilidades de interacción –gráficos, lenguaje corriente hablado y escrito, etc.– y deberá ser fácilmente conectable a las redes de telecomunicaciones. Para su consecución los japoneses han puesto en marcha un plan de investigación, de diez años de duración, que viene funcionando desde enero de 1982. En Estados Unidos este proyecto ha tenido su réplica en dos planes de investigación que, entre otras cosas, tienen por objetivo potenciar la investigación en inteligencia artificial. En Europa, el proyecto ESPRIT ha dado también un papel relevante a la inteligencia artificial. Desde la perspectiva actual estos proyectos responden a una necesidad de progreso bien sentida, pero el futuro depara siempre muchas sorpresas y será la experiencia la que juzgará la cantidad y la calidad de los resultados obtenidos hacia la inteligencia artificial.

J. Agustí
y J. J. Villanueva
Universidad Autónoma de Barcelona

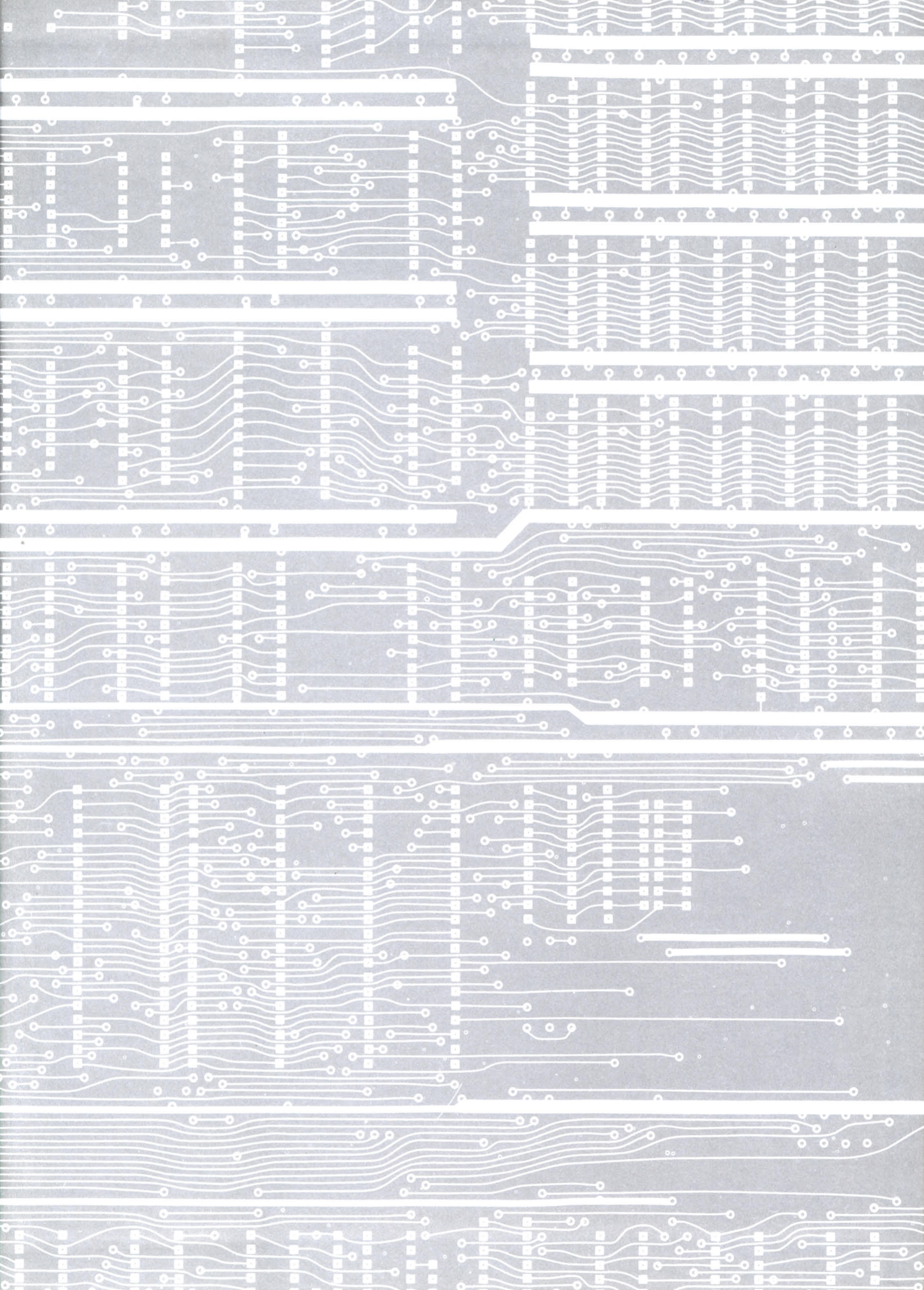
Basic Advanced — Indice

En la ilustración del Basic Advanced han colaborado: Marka-Zefa, Vincenzo Pirozzi, Patrizia Savarese y Savarese/Fradin.

Fe de erratas

Página	Errata	Corrección
1757	30. La matriz está (...)	1754. La matriz está (...)
1757	(...) puede verse en la página 32	(...) puede verse en la página 1756
1757	(...) flujo de las páginas 34 a 39	(...) flujo de las páginas 1758 a 1763
1760	grama de flujo (pág. 38) no	grama de flujo (pág. 1762) no
1792	4080 (...) ALLINEAMENTI (...)	4080 (...) ALINEADOS (...)
1797	grafía de la página 56, (...)	grafía de la página 1782, (...)
1798	(...) primera foto de la página 56.	(...) primera foto de la página 1782.
1802	na 57 sólo se habrá (...)	na 1783 sólo se habrá (...)
1802	(...) en las páginas 79 y	(...) en las páginas 1825 y
1802	80. Las funciones (...)	1826. Las funciones (...)
1805	ejemplo de la pág. 61 (...)	ejemplo de la página 1787 (...)
1813	12229 GOSUB 11000	12220 GOSUB 11000
1842	102 hay el campo (...)	1826 hay el campo (...)
1844	122 se ha representado el	1846 se ha representado el
1844	de la página 123.	de la página 1847.
1862	ENU":SL$="/"	ENU/:SL$="/"
1863	2345 GOTO 2345	*anular*
1868	ENU":SL$="/"	ENU/":SL$="/"
1869	2216 GOTO 2216: (...)	2216 GOTO (...)
1870	5340 F2(...)CHR(192))	5340 F2(...) "–")
1872	15552 G$ = "*V*":I'$ = "S":O$ = "*RWW*":LY = 18 (...)	G$ = CHR$(1) + CHR$(&H56): I$ = CHR$(1) + CHR$(&H53): O$ = CHR$(1) + CHR$(&H52): + "–": LY = 17: (...)
1872	15725 X = O: (...)	15725 X = 1: (...)
1873	15765 PRINT CHR$ (203)	15765 PRINT CHR$ (&HCC)
1883	La lógica del programa	Los listados/Versión Commodore 64
1890	Los listados/Versión Commodore 64	La lógica del programa
1894	171 y 172 no genera copias	1894 y 1895 no genera copias
1935	*Entre 5000 y 500Y*	*Intercalar:* 5002 POKE 52,64:POKE 56,64
1937	9150 REM ROUTINE ERRORE	95150 RUTINA ERROR
2251	(...) hasta un máximo de trece.	(...) hasta un máximo de dieciocho.
2278	*Tabla* Canguro 0 o 1	Canguro 1 o 1
2395	4650 RE = 1 + RE	4650 RE = 1 + RE: IFRE > 20 THENRETURN
2395	4700 R8 = R8 + 1:A$ = " ":FORI = 1TO10	4700 R8 = R8 + 1:IFR8 > 20 THENRETURN ELSEA$ = " ":FORI = 1TO10





813,192:E814,169:E815,1:E816,208:E817
328,3:E829,173:E830,128:EBK E831,192
18: "(LOADING INTEGER INTO LANGUAGE
ALDA--NO CARD OR CAN'

EDICIONES
FORUM





ENCICLOPEDIA DE LA INFORMATICA
MINIORDENADORES Y ORDENADORES PERSONALES